Date of acceptance Grade

Instructor

Protein Function Prediction using Biomedical Literature

Elaine Zosa

Helsinki April 12, 2017

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Elaine Zosa

Työn nimi — Arbetets titel — Title

Protein Function Prediction using Biomedical Literature

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
	April 12, 2017	72 pages + 0 appendices

Tiivistelmä — Referat — Abstract

Protein function prediction aims to identify the function of a given protein using, for example, sequence data, protein-protein interaction or evolutionary relationships. The use of biomedical literature to predict protein function, however, is a relatively under-studied topic given the vast amount of readily available data. This thesis explores the use of abstracts from biomedical literature to predict protein functions using the terms specified in the Gene Ontology (GO). The Gene Ontology (GO) is a standardised method of cataloguing protein functions where the functions are organised in a directed acyclic graph (DAG). The GO is composed of three separate ontologies: cellular component (CC), molecular function (MF) and biological process (BP).

Hierarchical classification is a classification method that assigns an instance to one or more classes where the classes are hierarchically related to each other, as in the GO. We build a hierarchical classifier that assigns GO terms to abstracts by training individual binary Naïve Bayes classifiers to recognise each GO term.

We present three different methods of mining abstracts from PubMed. Using these methods we assembled four datasets to train our classifiers. Each classifier is tested in three different ways: (a) in the paper-centric approach, we assign GO terms to a single abstract, (b) in the protein-centric approach, we assign GO terms to a concatenation of abstracts relating to single protein; and (c) the term-centric approach is a complement of the protein-centric approach where the goal is to assign proteins to a GO term. We evaluate the performance of our method using two evaluation metrics: maximum F-measure (F-max) and minimum semantic distance (S-min).

Our results show that the best dataset for training our classifier depends on the evaluation metric, the ontology and the proteins being annotated. We also find that there is a negative correlation between the F-max score of a GO term and its information content (IC) and a positive correlation between the F-max and the term's centrality in the DAG. Lastly we compare our method with GOstruct, the state-of-the-art literature-based protein annotation program. Our method outperforms GOstruct on human proteins, showing a significant improvement for the MF ontology.

ACM Computing Classification System (CCS):
J.3 [Life and Medical Sciences],
G.3 [Probability and Statistics],

Avainsanat — Nyckelord — Keywords
bioinformatics, protein annotation, literature mining, hierarchical classification

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

The goal of protein function prediction, also known as protein annotation, is to characterise a protein according to its biomedical function, the biological processes it is involved in and where it is located in the cell. Functional annotation is important in diverse areas of study such as drug development [DDB11], understanding the aetiology of human diseases [WVS+99] and development of genetically modified crops [VRH+87].

While we rely on laboratory experiments to identify the functions of uncharacterised proteins, such experimentation is a costly and time-intensive task. Consequently, the research community cannot keep up with the accelerating rate at which new proteins sequences are identified, making computational methods a more cost-effective means of inferring protein function [RCO+13]. Some of the data sources being used in computational annotation include sequence data, interaction data and evolutionary relationships [KTNKH15, JOC+16, RCO+13]. In this thesis, we want to investigate the use of biomedical literature in computational annotation because few existing methods take advantage of the vast amount of information contained in scientific literature databases such as PubMed. We use abstracts instead of full-text papers because abstracts are readily available and moreover, it has been shown that it contains the highest ratio of keywords than any other section in an article [SPIBA03].

There are different ways of describing protein function. In this thesis, we focus on the system established in the Gene Ontology (GO) where functions are related to other functions in a directed acyclic graph (DAG) structure [ABB+00]. We make use of a classification method known as *hierarchical classification* that is capable of handling classes structured in such a manner.

1.1 Hierarchical Classification

Hierarchical classification is a special case of structured classification where the classes are organised in a pre-defined hierarchical structure. Given an *instance* (i.e. a data point), the goal is to assign this instance to one or more nodes in the hierarchy. This differs from hierarchical clustering where the hierarchy is learned directly from the data. Hierarchical classification is a supervised learning method while clustering is unsupervised. Traditional multi-class and multi-label classifiers are referred to as flat classifiers. Hierarchical classifiers can be thought of as multi-class and multi-label classifiers where the labels have a hierarchical relationship [SJF11].

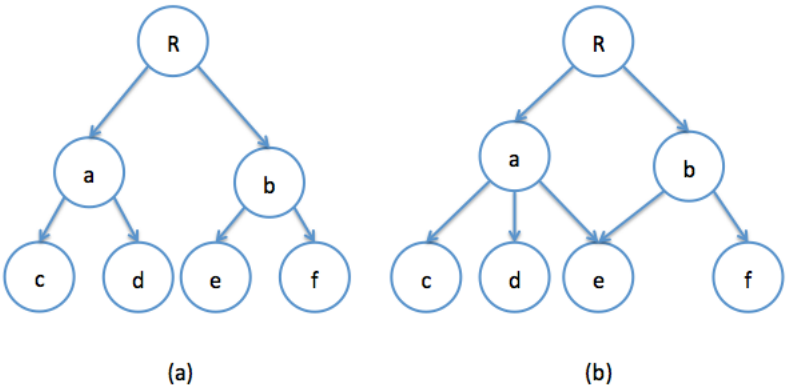

Figure 1: (a) tree (b) DAG

There are different kinds of hierarchical classification problems depending on the structure of the hierarchy being predicted, how the dataset is labelled and what is the expected prediction. Specifically, we classify them according to the following characteristics [SJF11]:

1. **Hierarchy structure**: The hierarchy can either be a tree or a non-tree directed acyclic graph (DAG). The difference between the two is that the nodes in a tree, excluding the root, have exactly one parent while the nodes in a DAG can have more than one parent (see Figure 1).

2. **Multiple-path labelling or single-path labelling**: Multiple-path labelling means that an instance can be assigned to more than one path to the root. Single-path labelling means that the expected prediction is a single path to the root.

3. **Full-depth labelling or partial-depth labelling**: Full-depth labelling means that an instance must be assigned to a one or more leaf nodes. Partial-depth labelling means that an instance can be assigned to any node. Whether a task requires full-depth or partial-depth labelling primarily depends on whether all instances in the data have leaf-level labels or not.

Protein function prediction is a hierarchical classification problem on a DAG that has multi-path labelling, because proteins can have more than one function, and has partial-depth labelling, because not all the labels in the dataset are leaf terms (in the GO, nodes are referred to as *terms*).

There are different methods of building a hierarchical classifier depending on the problem we want to solve. For this thesis, we choose a method called the *local classifier per node (LCN) approach*. In this approach, every node in our target hierarchy (in this case, the GO hierarchy) is considered a class and we build a base binary classifier for each class that is trained to recognise instances that belong to that class [SJF11].

Since each node is treated as an independent class and classifiers are trained independently, LCN is indifferent to the hierarchy structure. This independent training also means that this approach can handle multi-path labelling inherently because any number of classifiers can make a positive prediction for a test instance. Similarly, partial-depth labelling is also handled because classifiers from any level of the ontology can make a positive prediction. However, the LCN approach has drawbacks when it comes to single-path labelling and full-depth labelling problems. Single-path labelling means we are restricting our prediction for a test instance to a single path but since the classifiers are trained independently there is nothing that would stop classifiers that do not belong to the same path from making a positive prediction. With regards to full-depth labelling, we also cannot guarantee that at least one leaf-level classifier will make a positive prediction. There are several methods of handling these issues such as running the classifiers in a top-down manner [KS97] or using an approach known as binarized structured label learning [WZH05]. However these issues are outside the scope of this thesis.

Another factor that we need to consider for the LCN approach is dividing the training set into positive and negative instances for each node in the ontology. In Eisner (2005), the authors compared four data division strategies for protein function prediction: (a) *Exclusive*, (b) *Less Exclusive*, (c) *Less Inclusive*, and (d) *Inclusive* [EPS$^+$05] (see Figure 2). In the *Exclusive* strategy, the positive training set of the binary classifier for class c is composed only of training instances labelled with c and the rest of the training instances are in the negative training set. For the *Less Exclusive* approach, the positive set remains the same however training instances labelled with the descendants of c are removed from the negative set and discarded. In the *Less Inclusive* approach, the training set is composed of instances labelled with c or any descendants of c and the rest are in the negative set. In the *Inclusive* approach, the positive set remains the same while instances labelled with any ancestor of c are removed from the negative set and discarded. The study found that as the positive set becomes more inclusive, the F-measure increases. Based on their results, we selected the *Less Inclusive* data division strategy because it results in a

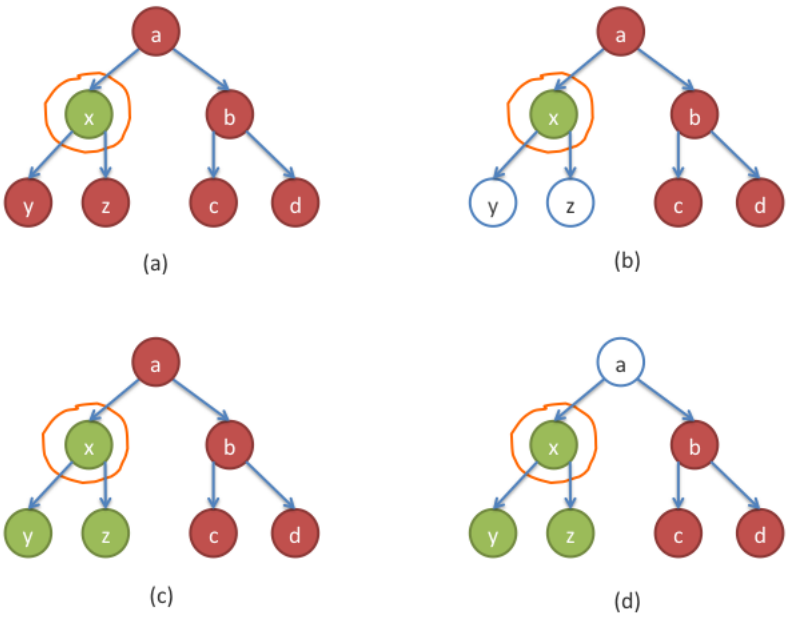

Figure 2: Strategies for dividing a dataset into positive and negative training sets to train a binary classifier for class x. Instances labelled with classes in green nodes are in the positive set, red nodes in the negative set and white nodes are discarded. These strategies are called: (a) Exclusive, (b) Less Exclusive, (c) Less Inclusive, and (d) Inclusive.

larger positive set and is simple to implement.

1.2 The Gene Ontology

The Gene Ontology (GO) is a set of terms used to catalogue protein functions and the relationships between them, organised in a directed acyclic graph (DAG). It integrates information from multiple databases that contain protein annotations for different model species [ABB+00, dPŠD11].

The GO is composed of three separate ontologies: *cellular component (CC)*, *molecular function (MF)* and *biological process (BP)*. CC contains the terms describing locations of proteins within the cell, MF encompasses the terms describing protein functionality and BP encompasses the biological processes that a protein is involved in. Inter-ontological relationships between terms are also possible, that is, a term from one ontology can have ancestors or descendants from another ontology. GO terms have a unique GO ID and a descriptive name. Relationships between GO terms are of different types:

- **Is-a**: This implies that a term is a more specific instance of its parent. An

example from MF is GO:0005515 (protein binding) being a child term of GO:0005488 (binding) since protein binding is a more specific function than binding. "Is-a" is the most common relation in the GO.

- **Part-of**: This is commonly found in CC and implies that a term is a component of its parent. An example is GO:0005737 (cytoplasm) as part of GO:0099568 (cytoplasmic region) since the cytoplasm is physically contained inside the cytoplasmic region.

For this thesis, we treat all relations as "is-a" relations, meaning that we consider terms to be more specific instances of their parents and more general instances of their children. When we trace a path from a term towards the root, the terms we encounter successively becomes more generic. Another implication of the "is-a" relation is that a protein that is labelled with a term is also labelled with the ancestors of that term. This is the *true path rule* of protein function annotation. Figure 3 shows a subgraph of the CC ontology ending in GO:0070013 (intracellular organelle lumen). In this subgraph we can see that the root's children are more generic (cell, organelle, etc) compared to to terms farther down (organelle lumen, intracellular organelle lumen, etc).

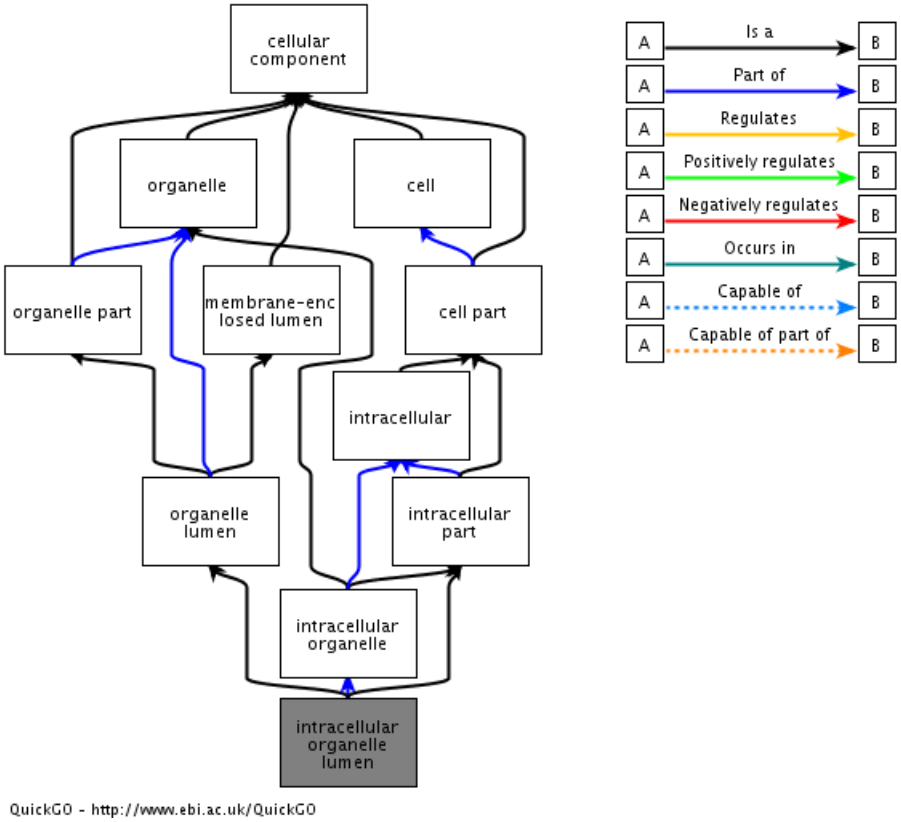

Figure 3: A subgraph of the CC ontology [BDH+09].

1.2.1 Gene Ontology Annotations

Proteins can be annotated with any number of GO terms. Each protein annotation comes with an *evidence code*: a two or three-letter code specifying on what basis an annotation was made. A protein annotated with the same GO term but with different evidence codes means that the annotation has been determined with multiple methods.

In general, annotations with experimental evidence codes are considered the most reliable because these are derived directly from experiments [dPŠD11]. Experimental evidence codes are EXP, IDA, IPI, IMP, IGI and IEP. Annotations derived from such methods as sequence similarity, sequence alignment and sequence modelling are computationally inferred annotations. Computational evidence codes are ISS, ISO, ISA, ISM, IGC and RCA. Annotations can also be made on the basis of an author's statement but without the original experimental evidence (TAS and NAS). Annotations can also be made on the basis of a curator's expert knowledge (IC and ND). Lastly, annotations assigned without human supervision have the evidence

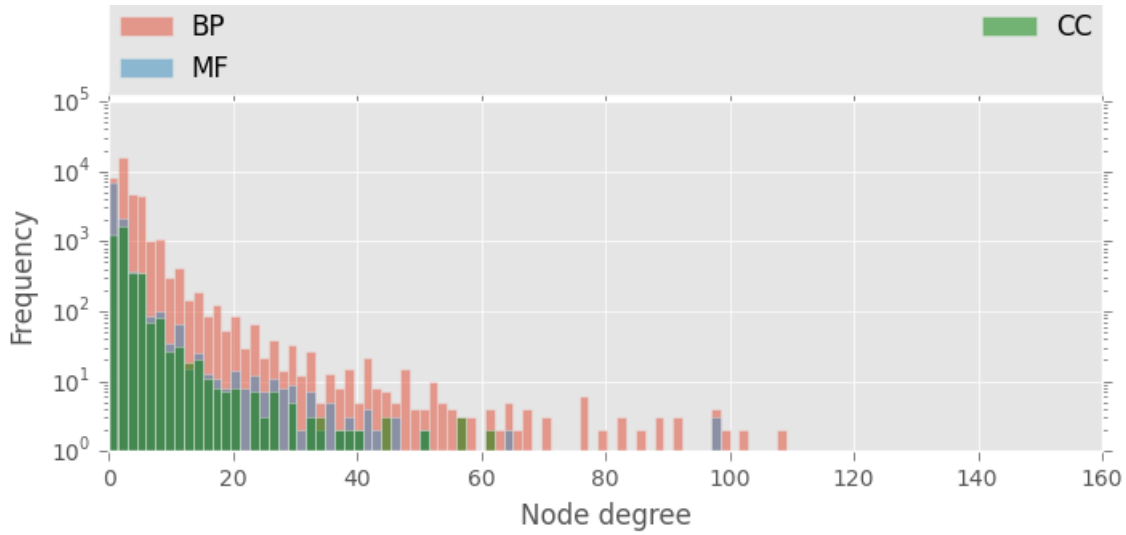

Figure 4: Degree histograms of cellular component (CC), molecular function (MF) and biological process (BP).

code IEA (inferred from electronic annotation) [dPŠD11].

IEA annotations make up a vast majority of GO annotations. Currently, only 0.8% of annotations are non-IEA. Because of the lack of human supervision in IEA annotations, many studies on computational inference methods do not make use of these annotations because it leads to a situation where models trained to infer annotations are trained on data generated by other models trained on similar data [dPŠD11, RSA+07]. However IEA annotations are not without their uses. It has been found that a majority of IEA annotations are eventually experimentally confirmed [dPŠD11]. For this thesis, we disregard all IEA annotations.

1.2.2 Analysis of the Gene Ontology

The GO ontologies are large and complex DAGs that can be difficult to visualise. The following analysis is to provide an idea of that complexity using measures from network studies such as node degree and node centrality.

The degree histogram in Figure 4 shows the distribution of a degree value in the GO. In a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, the degree of a node $v \in V$ is the number of edges connected to v. In a directed graph, this means the incoming and outgoing edges.

Since most GO terms are leaves (see Table 1), we expect them to have low de-

grees. This is evident in the histogram where the most high-frequency values are concentrated on the left side (Figure 4).

A majority of the nodes in CC have degrees of 1 or 2 which is expected since most nodes are leaf nodes and the average number of parents 1.9. The highest degree node is GO:0043234 (protein complex), a node two edges away from the root with one parent and 464 children giving it a degree of 465.

The degree histogram of MF closely resembles that of CC with most nodes having degrees of 1 or 2. As with CC, there is one outlier here and that is GO:0016616 (oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor), a node of degree 282 (one parent and 281 children) located four edges away from the root.

The degree histogram of BP shows that it is more connected than CC or MF. Most of the nodes have degrees 1 to 3 and this is the only ontology where nodes with degree 2 outnumber those with degree 1. BP also has one outlier node, GO:0044767 (single-organism developmental process), a node two edges away from the root with degree 420 (two parents and 418 children).

In a network or graph, there are some nodes that are more influential than others in the sense that they are more connected to the rest of the graph. This does not necessarily refer to the highest degree nodes because degree only refers to direct connections without taking into account the connections of that node's neighbours (their parents and children). In the GO, a more influential node could mean that is easier to assign. For this thesis, we will use centrality as the metric for influence [Bor05]. Since there are many kinds of centrality, we limit it to the following:

1. **Degree centrality**: for node v, the degree centrality is the fraction of nodes in the graph that v shares an edge with. For our ontologies, this refers to the nodes' parents and children divided by the total number of nodes in the ontology. The ontology is treated as an undirected graph. Since degree centrality is simply the degree of the node normalised by the total nodes in the graph, this will not give us any information we did not already know from the degree histogram.

2. **Closeness centrality**: for a node v, this is the sum of the shortest paths from node v to all other nodes in the graph, divided by the total number of

nodes, computed as [Bav50]:

$$C(v) = \frac{n-1}{\sum_{u=1}^{n-1} d(u,v)} \tag{1}$$

where n is the number of nodes in the graph. Higher values indicate greater centrality. Closeness centrality is a more informative than degree centrality since it takes into account the shortest distance from a node v to all other nodes in the graph.

3. **Betweenness centrality**: for a node v, the betweenness centrality is the number of times that v is on the shortest path between all pairs of nodes in the ontology normalised by the total number of shortest paths between all pairs of nodes, computed as [Fre77]:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{2}$$

where V is the set of nodes in the graph, $\sigma(s,t)$ is the number of shortest paths between nodes s and t and $\sigma(s,t|v)$ is the number of those paths that pass through v. Higher values indicate greater centrality. Betweenness centrality tells us how often v acts as a bridge between two other nodes in the graph.

Figures 5 to 7 shows the closeness centrality distribution for the CC, MF and BP ontologies, respectively.

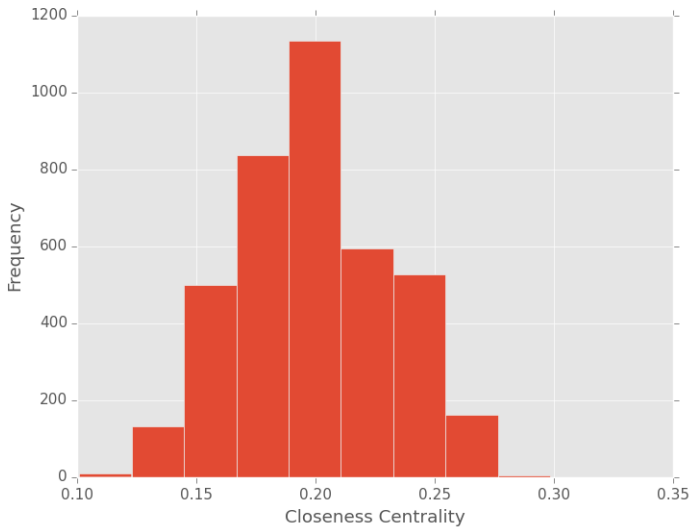

Figure 5: Closeness centrality histogram of cellular component (CC).

The node with the highest closeness centrality of 0.320 is GO:0043234 (protein complex), the same node with the highest degree in CC. The highest degree CC nodes also have some of the highest closeness centrality values.

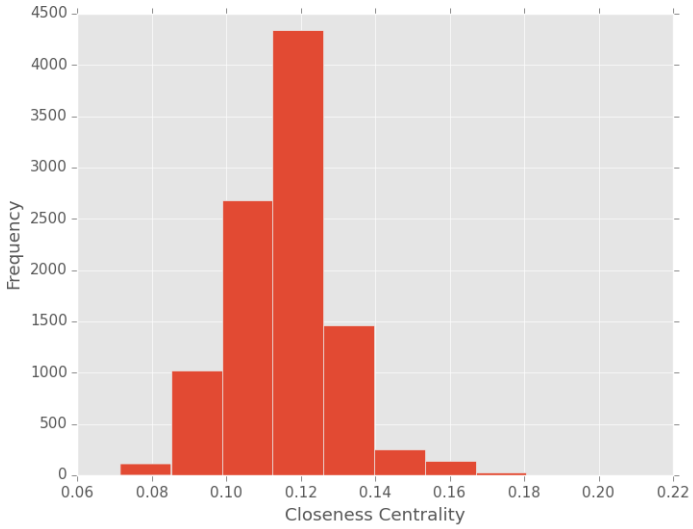

Figure 6: Closeness centrality histogram of molecular function (MF).

The most central node in MF in terms of closeness centrality is GO:0003824 (catalytic activity), a node two edges away from the root with a degree of 29 (one parent and 27 children). This is closely followed by the root node of MF, GO:0003674. An interesting observation is that the node with the highest degree does not even appear in the top 10 most central nodes. In general, MF nodes have lower centrality than CC nodes. This supports the observation that MF is less-connected than CC.

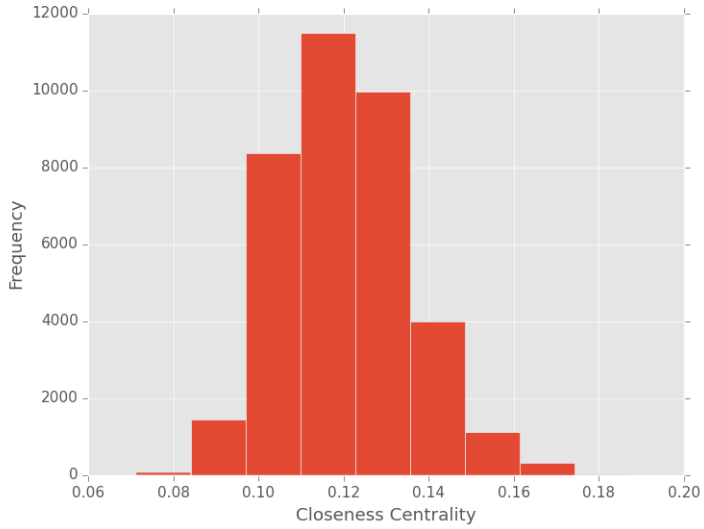

Figure 7: Closeness centrality histogram of biological process (BP).

In Figure 7, we observe that even though BP has more edges per node than CC, it turns out that in general, CC nodes are more central than BP nodes. The most central BP node is GO:0044763 (single-organism cellular process), a sibling of GO:0044767, the highest degree node in BP. Although GO:0044767 has almost twice the degree of GO:0044763 (420 versus 258), the former is not in the top tern most central nodes implying that its children are more isolated than the children of the latter.

Figures 8 to 10 shows the distribution of betweenness centrality values for the three ontologies. All of them show that most nodes don't act as bridges to other nodes which is expected since the majority of them are leaf nodes which by definition cannot act as bridges because they are terminal nodes.

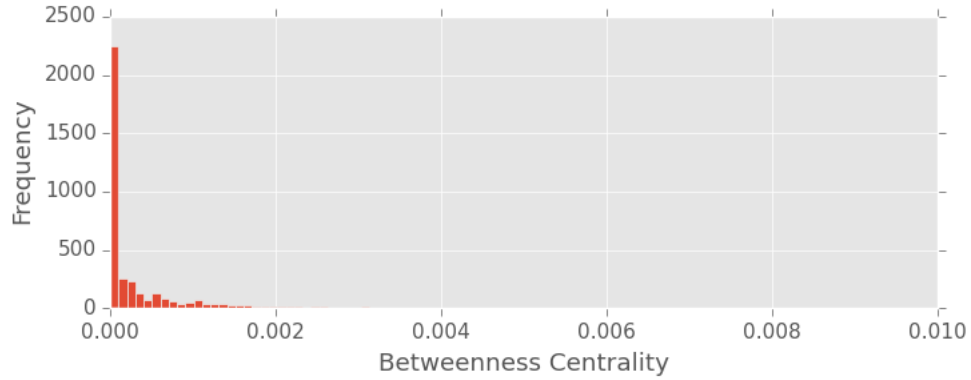

Figure 8: Betweenness centrality histogram for CC.

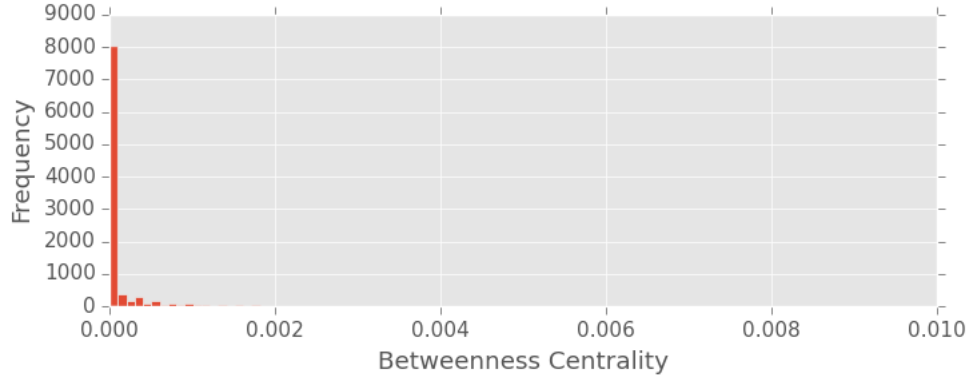

Figure 9: Betweenness centrality histogram for MF.

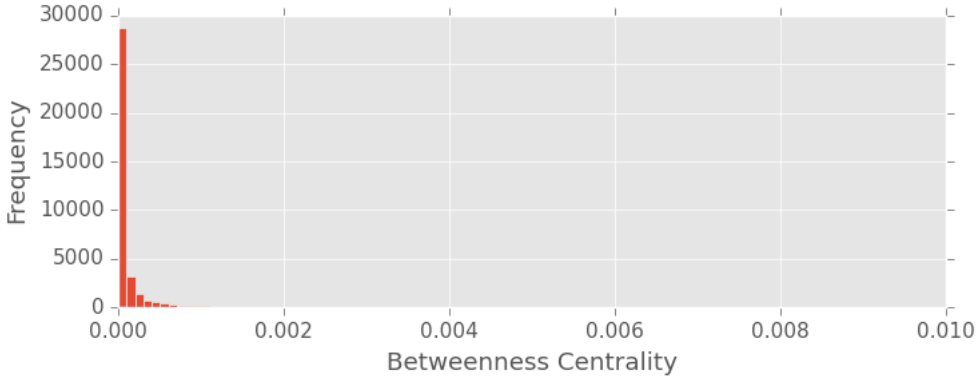

Figure 10: Betweenness centrality histogram for BP.

1.3 Related Work

In protein function prediction, some commonly used data sources include sequence data, interaction data (e.g. protein-protein interaction) and evolutionary relationships. An example of a protein function prediction method that uses sequence data is PANNZER (Protein Annotation by Z-score) [KTNKH15]. PANNZER has two main goals: predict GO annotations of an unknown protein and predict the text description of said protein. We will only discuss how PANNZER does the former since this thesis is only concerned with predicting GO terms. The process starts by running the sequence of the protein we wish to annotate against the BLAST database to get a list of proteins with similar sequences. Since there can be a large number of hits that turn out to be false positives, the results are filtered based on some sequence characteristics such as percent identity, alignment coverage and sequence length. From the filtered sequences, information such as the GO terms associated with each protein sequence and the size of the GO class (the number proteins have been annotated with that class) are collected. This information is then used in a regression model to score the GO classes of the filtered sequences and the GO classes with the top scores will be the predicted annotations of our query sequence.

Another protein annotation tool that uses sequence similarity is ARGOT (Annotation Retrieval of Gene Ontology Terms) [FCV+09]. Like PANNZER, ARGOT starts with a query of the uncharacterised protein sequence against the BLAST database. The GO annotations of the top BLAST hits are then used as the basis for the rest of algorithm (the number of hits to be analysed can be specified by the user). These classes are propagated upward to the root and then combined to form a trimmed GO graph. This graph is further refined by eliminating paths unlikely to be associated with the query by computing the Z-score of a path based on the weights of the node in that path. The weight of a node is calculated by summing the log of the BLAST e-value scores of its descendants. This means leaf nodes will have the weight only of its own BLAST e-value score. Paths that have a Z-score below a certain threshold are removed. Lastly, the remaining nodes are ranked based on their *information content (IC)* score, computed as:

$$IC = -log_2 p(c) \tag{3}$$

where $p(c)$ is the probability of GO term c or any of its descendants being used as an annotation of a protein in the dataset [Res95, CR13].

The IC score gives higher scores to rarer GO terms (usually, these are the more specialised terms located on the lower levels of the ontology) and lower scores to more generic GO terms that are close to the root. Root nodes will have an IC of zero because its descendants encompass the entire ontology.

Literature-based protein annotation methods are rarer than sequence-based methods. One such method is Text-KNN which uses abstracts to extract text features that are useful for differentiating GO classes from each other [WS13]. Text-KNN begins by collecting the abstracts of papers provided as evidence in the UniProt-GOA annotations and calculating a Z-score for each word in the collected abstracts for each GO class. The Z-score of a word w for a GO class c is computed as:

$$Z^w_{c,c'} = \frac{P(w|c) - P(w|c')}{\sqrt{\dot{P} \cdot (1 - \dot{P}) \cdot \left(\frac{1}{|D_c|} + \frac{1}{|D_{c'}|}\right)}} \tag{4}$$

$$\dot{P} = \frac{|D_c| \cdot P(w|c) + |D_{c'}| \cdot P(w|c')}{|D_c| + |D_{c'}|} \tag{5}$$

where c' refers to any GO class that is *not* c, D_c is the set of abstracts associated with class c, $D_{c'}$ is the set of abstracts *not* associated with class c and $P(w|c)$ is the conditional probability of word w appearing in an abstract belonging to D_c and conversely, $P(w|c')$ is the conditional probability of word w appearing in an abstract belonging to $D_{c'}$. A word w for a class c with a Z-score higher than a pre-determined threshold is selected as a characteristic word for that class.

During testing, a unseen protein p is represented as a n-dimensional vector where n is the total number of characteristic words for all classes. Each element x_i in the vector is a weight calculated by dividing the number of times that a characteristic word w_i appears in abstracts related to protein p by the total number of times all other characteristic words appear in abstracts related to protein p. To classify protein p, the cosine distances between the vector representing p and the vectors representing annotated proteins in the dataset are computed and using a k-nearest neighbour classifier (kNN), protein p is annotated with the GO annotations shared by at least three of the 10 nearest neighbours of p. In order to get enough characteristic features for each class, the researchers limited their classifier to 10 MF and 24 BP terms that are direct children of the root. The researchers tested their classifier using the term-centric approach, meaning that precision, recall and F-measure were calculated for each of the 10 MF and 24 BP terms. A later study from the same group replaced the kNN classifier with a soft-margin support vector machine (SVM) that outputs a probability estimate of an instance belonging to each class [SBW15].

Another literature-based method that is closer to our goal for this thesis is GOstruct, a literature-based protein function method that uses a structured output SVM to predict labels that are consistent with the GO structure [FKBHV15]. Structured output SVM is an example of a global hierarchical classifier that uses kernel methods to handle hierarchical output spaces [SBH10]. The GOstruct data is composed of 13,530,032 abstracts and 595,100 full-text papers from the PubMed Open Access Collection. Two types of features are used: co-mentions and bag-of-words. Co-mentions are co-occurences of a protein and a GO term in literature. This co-occurence can be in the same sentence or in the same paragraph or abstract. Bag-of-words are extracted from sentences that mention a protein.

For training, GOstruct is provided with the set of literature features and the gold standard annotations from the Gene Ontology Annotation (GOA) database. Then the structured output SVM learns to associate the given features with GO annotations. For testing, a set of literature features pertaining to a protein are provided and GOstruct predicts the labels (GO terms). The classifier is limited predicting GO classes that have at least 10 proteins associated with it and only human and yeast proteins are used in the GOstruct dataset. In Section 4, we compare GOstruct performance with the results from our own method.

2 Methods

2.1 The Naïve Bayes Classifier

The Bayesian classifier is a supervised method that assigns an instance $X = \{x_1, x_2, .., x_n\}$ to class a $Y = y$, where X is a *feature vector* and Y is a *class variable*, with the probability given by Bayes' theorem:

$$P(Y = y|x_1, x_2, ..., x_n) = \frac{P(y)P(x_1, x_2, ..., x_n|y)}{P(x_1, x_2, ..., x_n)} \tag{6}$$

where $P(y)$ is the fraction of the training data that belongs to class y and $P(x_1, x_2, ..., x_n|y)$ is the *likelihood* of this instance of X given class y. $P(x_1, x_2, ..., x_n)$ is the marginal probability of this instance.

In a binary classification task where Y can be one of two classes, say positive class (1) and negative class (0), it is sufficient to maximise the numerator in Equation 6 because the denominator is constant for all values of Y. More concretely, an instance

of X is assigned to the positive class if the following condition holds, otherwise it is assigned to the negative class:

$$\frac{P(Y = 1|x_1, x_2, ..., x_n)}{P(Y = 0|x_1, x_2, ..., x_n)} \geq 1 \tag{7}$$

Naïve Bayes (NB) assumes that the features $x_1, x_2, ..., x_n$ are independent of each other which is rarely true in real-world classification problems. This is known as the "naïve" assumption. Despite this assumption being violated in most problems we are interested in, NB is still a high-performing classifier particularly in text classification. Quinlan (2014) showed that NB performance is on par with decision tree algorithms [Qui14]. Domingos (1997) proposed that the explanation for NB's surprisingly good performance is the zero-one loss function which does not penalise classifiers on their probability estimates as long as it assigns the test instance to the correct class [DP97]. For instance, suppose a NB classifier estimates that an instance belongs to the positive class with 0.55 probability and to the negative class with 0.45 probability but even if the actual probability is 0.80 positive and 0.20 negative, the classifier would still output the correct classification (positive). For this reason, NB is known to be a good predictor but a bad probability estimator therefore we should not place too much confidence in the probabilities given by an NB classifier. Zhang (2004) proposes that another reason for NB's performance is that even though the features are dependent on each other, the dependence is evenly distributed in each class and furthermore, even if two features are dependent on each other, the dependencies may cancel each other out across all features [Zha04].

For this thesis, we chose Naïve Bayes as our base classifier because it is relatively fast to train, known to perform well in text classification and gives probability estimates (as long as we regard those probabilities with caution).

2.2 Datasets

2.2.1 GO Version

The GO is dynamic, which means terms and relations are constantly being added, modified or rendered obsolete. For instance, there were 3,000 terms in 2000 and it increased to more than 30,000 by 2010 [dPŠD11]. For this thesis, we use the GO version from April 2016 which has a total of 50,804 terms: 3,902 for cellular component (CC), 10,057 for molecular function (MF) and 36,854 for biological process

```
A0A023GUT0      LPN             GO:0005615      PMID:24823393   IDA             C       Leptin
A0A023GUT0      LPN             GO:0042517      PMID:24823393   IMP             P       Leptin
A0A023GUT0      LPN             GO:1990460      PMID:24823393   IDA             F       Leptin
```

Figure 11: Excerpt from the UniProt-GOA annotation file. This shows the protein Leptin (Accession ID: A0A23GUT0) annotated with three GO terms: (1) GO:0005615 from CC ontology supported by evidence code IDA; (2) GO:0042517 from BP ontology supported by evidence code IMP; and (3) GO:1990460 from MF ontology supported by evidence code IDA. The basis for all three annotations can be found in the same paper (PMID: 24823393).

(BP) [C+15]. The vast majority of GO terms are leaf terms (see Table 1).

Ontology	Terms	Relations	Leaves
CC	3,902	7,333	2,583
MF	10,057	12,282	8,001
BP	36,854	72,212	28,554

Table 1: Number of terms, relations and leaves in the GO version (April 2016) used for this thesis.

2.2.2 Gold Standard Annotations

Protein annotations are taken from the UniProt Gene Ontology Annotations (UniProt-GOA) database [HSMM+15]. The UniProt-GOA database is a database of manual and electronic protein annotations provided by research groups in the GO Consortium [C+15]. For this thesis, we use annotations of multi-species proteins, human proteins (*Homo sapiens*) and yeast proteins (*Saccharomyces cerevisiae*). The UniProt-GOA annotations includes a reference to the source of the annotation. If the supporting evidence has been published, this would contain the PubMed ID (PMID) of the paper discussing that evidence. We only select annotations with a supporting PMID and a non-IEA evidence code. We also remove GO:0005515 (protein binding) annotations because its over-abundance in the dataset could skew our results (it makes up 8.68% of of the total annotations and 32.48% of the MF annotations; see Figure 13). These annotations comprise our *gold standard* annotations because they have been reviewed by human experts. The annotations used for this thesis was downloaded from GOA on October 2016. Figure 11 is an excerpt from the annotation file. Table 2 shows a breakdown of the number of annotations (excluding GO:0005515) and proteins per species and ontology.

	CC	MF	BP	Total	Proteins annotated
Multi-species	197,829	146,609	397,455	741,893	126,648
Human	108,226	43,647	88,934	240,807	12,836
Yeast	13,763	13,571	22,806	50,140	6,624

Table 2: Breakdown of the number of gold standard annotations in our dataset (excluding GO:0005515).

In protein annotation, terms in the upper levels of the ontology are more likely to be used as annotations because they are more general and therefore easier to assign. This is shown in Figures 12 to 14 where terms at the third and fourth levels of the ontology contribute to the majority of gold standard annotations. For this thesis, we define the *level* of a term as the number of edges of the shortest path from the root to that term. For instance, in Figure 3, organelle lumen is on level 2.

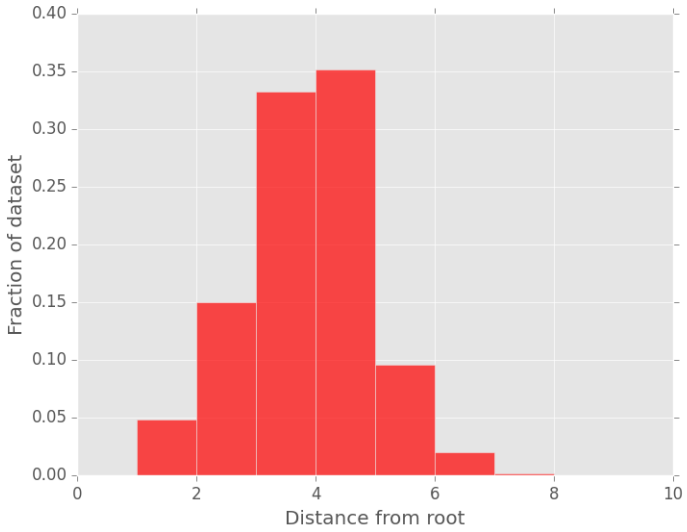

Figure 12: Fraction of CC multi-species gold standard annotations per level.

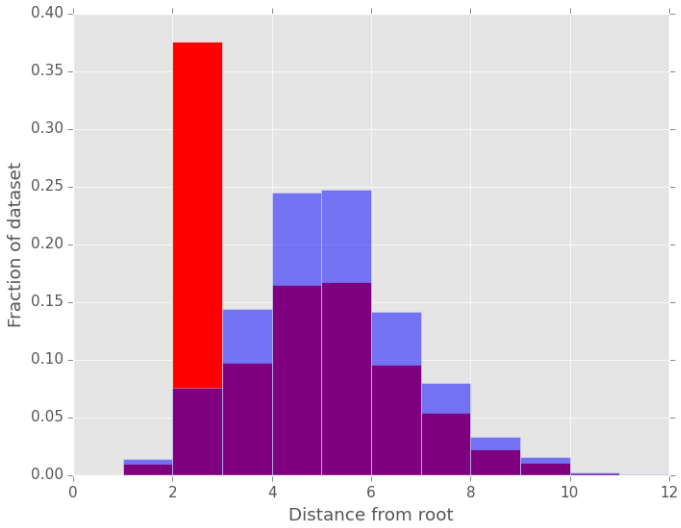

Figure 13: Fraction of all MF multi-species gold standard annotations per level including GO:0005515 (in red); excluding GO:0005515 (in purple).

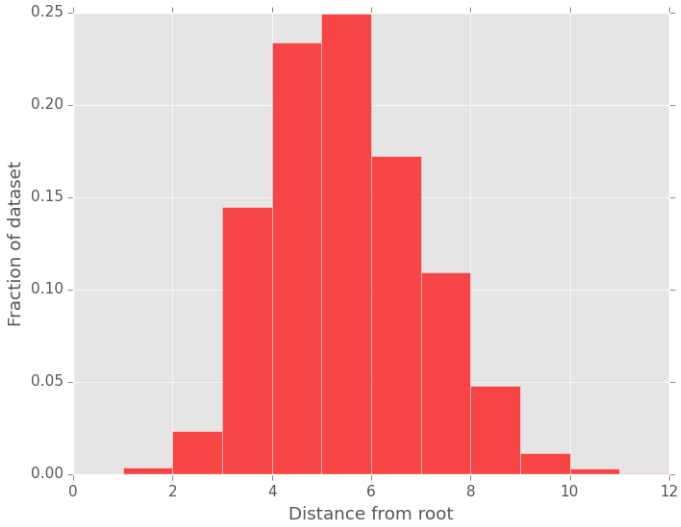

Figure 14: Fraction of BP multi-species gold standard annotations per level.

However the level of a term is not the only factor that explains its specificity because the most specific terms are leaf terms and they can be found at every level of the ontology. The *information content (IC)* of a term (Equation 3) is a measure of the bits of information added to a protein annotated with that term [CR13]. Figure 15 shows the relationship between a term's IC and its distance from the root. Notice

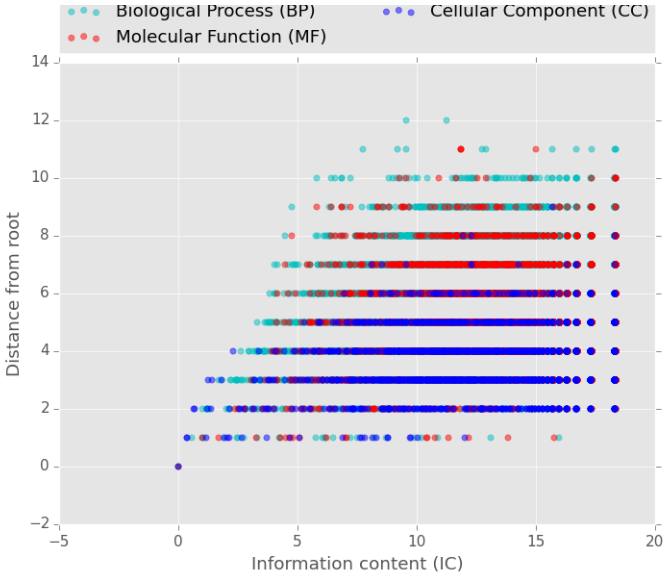

Figure 15: Relationship between the IC of a term and its distance from the root.

that the roots of the three ontologies (at distance 0) has an IC of 0 because no information is gained when a protein is annotated with a root term.

Next, we need to collect the abstracts of biomedical articles from which we will extract the text features to train and test our classifier. We have come up with three methods to associate abstracts with GO annotations and assembled four datasets from these methods. We refer to the resulting datasets as: (1) Curated, (2) PubMed GO, (3) PubMed Gene, and (4) Curated+PubMed Gene.

2.2.3 Curated Dataset

This dataset is composed of the abstracts of papers used to support an annotation and the abstracts will be labelled with those GO annotations. For instance, from Figure 11, the abstract for PMID 23823393 will be labelled with GO:0005615, GO:0042517, GO:1990460. Since these annotations have been assigned and reviewed by human experts, we can be assured that these abstracts are indeed relevant to the annotations they are associated with. For this reason, all our test instances will be taken from this dataset.

2.2.4 PubMed GO

This dataset is assembled by mining PubMed for papers relevant to the names of GO terms. That is, we query PubMed using the GO names (or its synonyms) as the search keywords and the returned abstracts will be labelled with that GO term. For disk space and computational time reasons, we only save at most 10 results.

For instance, the name of GO:0005615 is "extracellular space". We query PubMed for papers related to "extracellular space" and save the abstracts of the first 10 papers returned by PubMed. These abstracts will be labelled with GO:0005615. Figure 16 shows an illustration of this process.

The rationale behind this is that papers that talk about protein functions by mentioning GO terms are likely to be similar to papers about proteins that associate them with their functions.

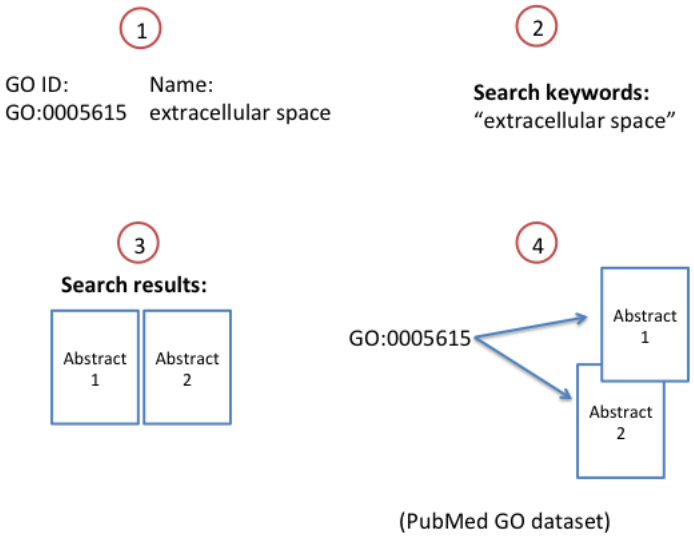

Figure 16: Overview of assembling the PubMed GO dataset. In this example, we want to find abstracts related to GO:0005615 (extracellular space).

2.2.5 PubMed Gene

This dataset is assembled by mining PubMed for relevant papers using the names of annotated proteins. The abstracts relevant to a protein name will then be labelled with the GO terms associated with that protein. As with the PubMed GO dataset, we only save at most 10 abstracts per protein. Since proteins can be associated with multiple GO terms, the same abstract can also be labelled with multiple GO terms.

For instance, the protein Leptin has three GO annotations. We query PubMed for papers related to "Leptin", save at most 10 abstracts and label each of these abstracts with the three GO terms associated with Leptin. Figure 17 shows an illustration of this process.

The rationale behind this is more straightforward than PubMed GO. Since we are searching for papers that talk about a protein and label these papers with the protein's annotations, we can use this dataset to train a classifier to associate a protein's description with GO terms.

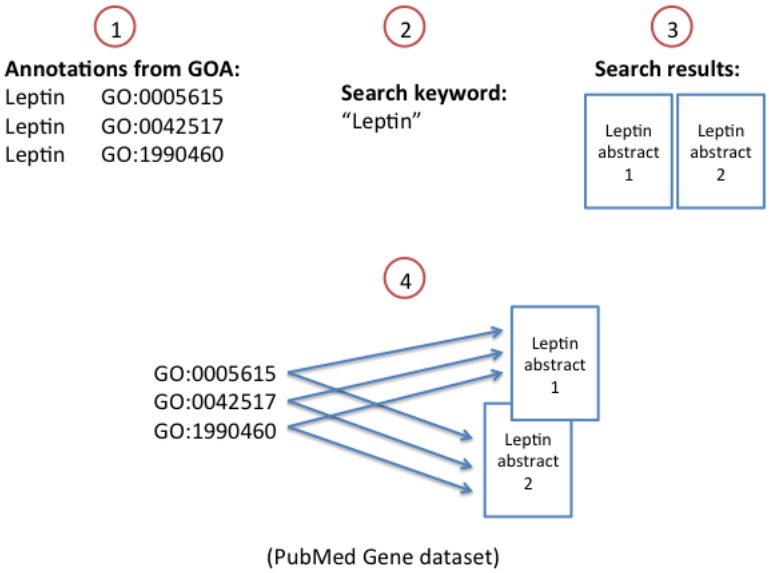

Figure 17: Overview of assembling the PubMed Gene dataset. In this example, the protein Leptin has three gold standard GO annotations. We search for abstracts related to "Leptin" in PubMed and associate the results (in this case, two abstracts) with the three GO annotations.

For the PubMed GO and PubMed Gene datasets, the labels associated with the abstracts are only inferred and have not been validated by human experts therefore we cannot use them as test instances in our experiments.

2.2.6 Curated+PubMed Gene

This dataset is simply a union of the Curated and PubMed Gene datasets. We do not include a Curated+PubMed GO dataset because initial experiments show it does not perform any better than the Curated dataset.

Table 3 shows the number of abstracts in each dataset.

Dataset	No. of Abstracts		
	CC	MF	BP
Curated	48,829	61,578	98,634
PubMed GO	15,627	51,941	104,750
PubMed Gene	161,802	161,027	184,102
Curated+PubMed Gene	170,735	168,633	282,476

Table 3: Datasets summary

2.3 Experimental Setup

Figure 18 shows an overview of our experimental setup. Each step will be covered in the subsequent sections.

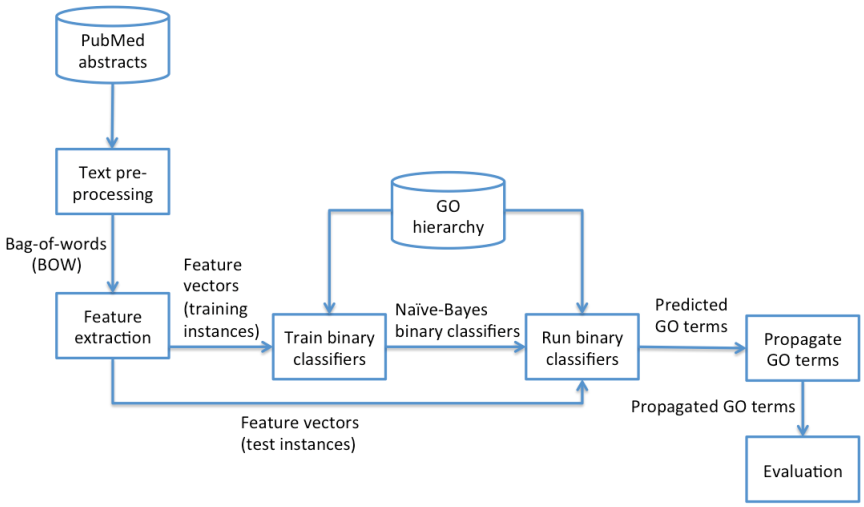

Figure 18: Overview of the protein function prediction experimental setup.

2.3.1 Text Preprocessing

In this step, we pre-process the abstracts in order to simplify the feature extraction phase. Specifically, we perform the following steps on all abstracts in our datasets:

1. Convert all characters to lowercase.
2. Remove all punctuation marks.
3. Remove English stop-words (common words in English) [Bir06].

4. Apply the Porter stemmer to all words. Stemming a word means reducing it to its root form [Bir06].

2.3.2 Feature Representation

After text preprocessing, the abstracts are decomposed into a bag-of-words (BOW). This means that an abstract is represented by the words it contains while disregarding the ordering of the words. The BOW representation simplifies the succeeding steps because it assumes that all words are independent of each other.

Next, we convert the BOW representation into feature vectors. In text classification, the most common feature vector is a count vector where each document is represented as a vector of the same size as the vocabulary size of the dataset and each element in the vector is the *term frequency*, the number of times a word (or term, in the information retrieval sense and not a GO term) appears in the document [MRS+08]. For this thesis, instead of using the term frequency we use the *term frequency-inverse document frequency (TF-IDF)*. The TF-IDF of a term is defined as:

$$TF\text{-}IDF(term) = TF(term) \times IDF(DF, term) \tag{8}$$

$$IDF = \log\left(\frac{n}{DF}\right) + 1 \tag{9}$$

where TF is the term frequency and DF is the *document frequency*, the number of documents where the term appears and n is the number of documents in the dataset. The effect of the TF-IDF transformation of count vectors is to give less weight to frequent terms that are encountered in many documents in the dataset and give more weight to rarer terms because the former has less discriminative ability than the latter [MRS+08].

2.3.3 Training the Classifiers

We build a hierarchical classifier using the *local classifier per node (LCN)* approach with the *Less Inclusive* dataset division strategy. This means we train a binary Naïve Bayes classifier for each GO term and that the positive training set for a term c is composed of the training instances labelled with c or any descendants of c while the negative set is composed of instances that are not in the positive set. Since

there will be terms for which their positive set will be composed of very few, or zero, instances, we impose a rule that we only train a classifier if the positive training set has at least five instances. This number is selected empirically.

The Naïve Bayes algorithm has a prior that is intended to reflect the background probability of a class being the true class. This prior can be estimated from the training set or we can assume that both positive and negative classes are equally likely (equiprobable classes) and ignore the prior entirely. After initial experimentation, we decided to choose the latter strategy because it produces better results.

2.3.4 Testing Approaches

Testing a protein function prediction method can be viewed from three different perspectives [JOC+16]:

- **Paper-centric approach**: This approach addresses the question, *Given a paper (or an abstract), what GO terms are associated with it?* This means that we want to know what GO terms are discussed in a given abstract. This is only used for literature-based methods.
- **Protein-centric approach**: This approach addresses the question, *Given a protein, what GO terms are associated with it?* This directly addresses the purpose of protein function prediction, associating proteins with GO terms.
- **Term-centric approach**: This addresses the question, *Given a GO term, what proteins are associated with that term?* This is a complement to the protein-centric approach where instead of looking at a protein and its GO terms, we look at a GO term and the proteins associated with it.

2.3.5 Cross-Validation

In the case of the Curated dataset which we use in both training and testing, we want to be able to test all the instances in the dataset but also avoid overfitting our model. Overfitting is when a classifier achieves very high performance when it is tested on the same data it was trained on but performs poorly on unseen examples, thus, giving a very inaccurate measure of a classifier's performance [JWHT13, Fla12].

To avoid overfitting, we use *k-fold cross-validation*. We partition our dataset into k parts and for each partition, we train a classifier on the dataset without that

partition and test the resulting model on the partition we excluded and compute the performance metrics. We repeat this process k times and average the metrics [JWHT13, Fla12].

We apply 5-fold cross-validation only to the datasets that are derived from the Curated dataset: Curated and Curated+PubMed Gene. For the PubMed GO and PubMed Gene datasets, we remove instances from the training set that appear in the test set so that there is no intersection between the training and test data.

2.3.6 Evaluation Metrics

In binary classification, *precision*, *recall* and *F-measure* are common performance metrics [JWHT13, Fla12]. Precision refers to the number of true positives divided by the number of test instances predicted positive. Recall is the number of true positives divided by the number of test instances that actually belong to the positive class. F-measure is the harmonic mean between precision and recall.

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

$$F\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{12}$$

where TP means *true positive*, FP means *false positive* and FN means *false negative*.

We adapt precision, recall and F-measure in a hierarchical classification setting where predicted labels and true labels are sets of labels (instead of a single value) and these labels are hierarchically related. This means that predicted labels are not evaluated only as correct or incorrect but on their distance in the hierarchy from the correct labels. For example, consider the hierarchy in Figure 1b and a test instance whose true class is c and suppose that the classifier predicted c. This is considered a perfect prediction. Now suppose the classifier predicted d instead of c, this is not a perfect prediction but it is still close to the truth since they share a parent. The farther the prediction is from the true label, the less correct is becomes. This is also connected to the true path rule where a protein annotated with a term is also annotated with that term's ancestors. In the same manner, if we predict that a protein is associated with a term then we are also predicting that it is associated

with that term's ancestors. Therefore before evaluating the predicted annotations of a test instance, we propagate its set of true annotations and predicted annotations towards the root. Propagation means that all ancestors of a term are included in the set of annotations. This leads to the following equations:

$$TP_{hier} = |T \cap P| \tag{13}$$

$$FP_{hier} = |P - T| \tag{14}$$

$$FN_{hier} = |T - P| \tag{15}$$

where P is the set of predicted labels and their ancestors and T is the set of true labels and their ancestors.

We define *hierarchical precision*, *hierarchical recall* and *hierarchical F-measure* [RCO+13]:

$$precision_{hier} = \frac{TP_{hier}}{TP_{hier} + FP_{hier}} \tag{16}$$

$$recall_{hier} = \frac{TP_{hier}}{TP_{hier} + FN_{hier}} \tag{17}$$

$$F\text{-}measure_{hier} = \frac{2 \times precision_{hier} \times recall_{hier}}{precision_{hier} + recall_{hier}} \tag{18}$$

These are fair metrics because the closer terms are to each other in the hierarchy, the more ancestors they will have in common. One drawback of these metrics is that it leads to the shallow annotation problem, referring to the fact that since generic annotations are given the same weight as more specific annotations despite the difference in their informative content, proteins are more likely to be annotated with generic terms [dPŠD11].

An evaluation measure for ontological annotations known as *semantic distance* addresses the issue of shallow annotations [CR13]. Semantic distance takes into account the information content (IC) of the nodes in the ontology using the notions of *remaining uncertainty (ru)*, the sum of the IC's of the true annotations that has not been predicted by the classifier, and *misinformation (mi)*, the sum of the IC's of the incorrect annotations predicted by the classifier.

$$ru(T, P) = \sum_{v \in T - P} IC(v) \tag{19}$$

$$mi(T, P) = \sum_{v \in P - T} IC(v) \tag{20}$$

where T is the set of true annotations and P is the set of predicted annotations and IC is computed as in Equation 3.

Finally, the semantic distance is defined as the distance from the origin to the curve defined by ru and mi:

$$S_k = [ru^k + mi^k]^{\frac{1}{k}} \tag{21}$$

It is recommended to set $k = 2$ (Euclidean distance) because it penalises unbalanced predictions [CR13]. It is easy to see that for perfect predictions, remaining uncertainty, misinformation, and semantic distance are all equal to zero.

2.3.7 Thresholding

A Naïve Bayes classifier is probabilistic therefore it outputs the probability estimate of an instance belonging to a particular class. This means that the result of the testing phase is a matrix of probabilities $A^{n \times m}$ where n is the number of test instances and m is the number of classes. We want a decision rule $D(A) : \mathbb{R}^{n \times m} \to \{0, 1\}^{n \times m}$ which maps a real-valued probability estimate to one of the two classes, positive (1) or negative (0). This means that we need a probability threshold, known as our *decision threshold*, over which we assign instances to the positive class and all others

to the negative class. Naturally, we want to find the decision threshold which gives us the best performance as defined by our evaluation metrics.

First we want the decision threshold τ_1 that gives us the maximum F-measure or *F-max* [RCO+13, JOC+16]. F-max is defined as:

$$F\text{-}max = max_{\tau_1} \left\{ \frac{2 \times prec(\tau_1) \times rec(\tau_1)}{prec(\tau_1) + rec(\tau_1)} \right\} \tag{22}$$

To find this threshold, we calculate the F-measure for each threshold from 0.0 to 1.0 in increments of 0.01.

Next, we want the decision threshold τ_2 that gives the minimum semantic distance or *S-min* [CR13, JOC+16]. S-min with Euclidean distance is defined as:

$$S\text{-}min = min_{\tau_2} \sqrt{(ru^2(\tau_2) + mi^2(\tau_2)} \tag{23}$$

As with F-max, we also calculate semantic distance for each threshold from 0.0 to 1.0 in increments of 0.01. The decision thresholds τ_1 and τ_2 that gives the F-max and S-min, respectively, can be different from each other.

We will compute F-max for the paper-centric, protein-centric and term-centric approaches and S-min for the protein-centric approach.

2.4 Implementation

Most of the algorithms used in this thesis are well-known and have a number of implementations that have already been extensively tested and documented. For our binary Naïve Bayes classifier, we use the implementation provided by Scikit-learn [PVG+11]. Scikit-learn is a machine learning library written in Python that provides many widely-used classification algorithms, feature selection methods, evaluation methods and others. We also Scikit-learn's TF-IDF vectoriser to extract the TF-IDF feature vectors.

Natural Language Toolkit (NLTK) is a Python library for natural language processing tools and algorithms [Bir06]. We use NLTK to stem words and remove stop words in the text pre-processing phase.

Networkx is Python library of network analysis tools and other graph theoretical algorithms [SS08]. We use Networkx to compute centrality measures of GO terms and compute the degree histogram of the GO ontologies.

Entrez is a collection of data retrieval tools used to access NCBI databases including PubMed. We use Entrez to mine the PubMed database and download abstracts of papers. The Entrez API is part of the BioPython library [CAC+09].

Matplotlib is a plotting library that generates graphs such as histograms, scatter plots, etc [Hun07]. We use Matplotlib to visualise some of our results and analyse the data used in this thesis .

The data collection, training, testing, evaluation and data analysis modules are written by the author in Python 2.7.

3 Results and Discussion

3.1 Paper-centric Approach

In Table 4, we have the F-max scores of each dataset for the paper-centric testing approach. As previously mentioned, Curated and Curated+PubMed Gene are tested using 5-fold cross-validation and the reported numbers are the averages of the five runs.

In CC, the dataset with the highest F-max is PubMed Gene followed by Curated, PubMed GO and lastly, Curated+PubMed Gene. Curated+PubMed Gene's low F-max score can be attributed to its poor precision. While all other datasets have precision above 0.60, Curated+PubMed Gene has 0.435. In MF, Curated dataset has the highest F-max by a large margin. This followed by PubMed Gene, PubMed GO then Curated+PubMed Gene. As in CC, the reason for Curated+PubMed Gene's low F-max is because of its very low precision of 0.349. In BP, the ranking of datasets is similar to CC: PubMed Gene has the best F-max followed by Curated, PubMed GO and Curated+PubMed Gene.

For all four datasets, CC by far has the best F-max followed by MF then BP (0.593, 0.430 and 0.296, respectively). One of the reasons for this could be that the names of CC terms (e.g. names of cell parts) are easier to find "as-is" in literature than the more complex names of MF and BP terms (e.g. "ethylbenzene hydroxylase activity" or "establishment of nucleus localization"). Another reason is that CC is

Cellular Component (CC)				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.572	0.605	0.543	0.09
PubMed GO	0.541	0.690	0.445	0.42
PubMed Gene	0.583	0.632	0.541	0.17
Curated+PubMed Gene	0.483	0.435	0.543	0.45
Molecular Function (MF)				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.430	0.564	0.349	0.01
PubMed GO	0.332	0.566	0.234	0.01
PubMed Gene	0.360	0.487	0.285	0.01
Curated+PubMed Gene	0.323	0.349	0.302	0.34
Biological Process (BP)				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.287	0.318	0.261	0.02
PubMed GO	0.282	0.448	0.206	0.01
PubMed Gene	0.296	0.307	0.285	0.07
Curated+PubMed Gene	0.251	0.251	0.252	0.46

Table 4: Paper-centric results for all ontologies. The best-performing dataset for CC and BP is PubMed Gene and Curated for MF. Threshold τ_1 is the decision threshold at which we get our F-max.

the smallest ontology with 3,902 terms and 7,333 relations. Having a smaller but denser network means that two terms are more likely to be related to each other and more likely to have common ancestors. Since our performance metric is based primarily on the number of common terms between the predicted and true label sets, this leads to a higher F-max score.

An important observation is that the performance of the PubMed Gene and Curated datasets for CC and BP are very close to each other. This means that in literature-based protein function prediction, we can use a dataset with inferred labels and get results on par with a human-curated dataset. Given our simple method of assembling the PubMed Gene dataset and comparing it to the time and effort required for human-curated datasets, this result is particularly encouraging. As for MF, we find that it has the smallest PubMed Gene dataset with 161,027 abstracts (Table 3). Moreover when we consider only the *unique* abstracts (abstracts not found in the

other datasets), the number becomes even smaller which helps explain why it does not perform as well as the Curated dataset.

The worst performing dataset is PubMed Gene+Curated. This is rather surprising since the dataset with the best F-max score is either PubMed Gene or Curated and PubMed Gene+Curated is just a union of those two datasets. However we observe that unlike other datasets where precision is always higher than recall, in PubMed Gene+Curated recall is higher in CC and precision is only a little higher than recall in MF. Also notice that Curated+PubMed Gene has the highest decision thresholds (τ_1) in any ontology. All of this suggests that this semi-curated dataset causes the classifier to over-predict because of the larger size of the positive training set, resulting in more false positives (low precision).

The next worst performing dataset is PubMed GO where the low F-max score can be attributed to poor recall. We think that the explanation for this behaviour is in the size of this dataset. PubMed GO has the least number of abstracts for CC and MF. On BP where it is bigger than the Curated dataset, it still has the fewest number of unique abstracts. In fact, this dataset has the smallest percentage of unique abstracts (see Table 5). This means that the size of the positive training set is smaller compared to the other datasets which leads to more conservative predictions and thus, more false negatives (low recall).

	CC	**MF**	**BP**
Curated	83.75%	83.59%	83.63%
PubMed GO	67.29%	70.07%	75.41%
PubMed Gene	96.84%	86.42%	80.78%

Table 5: Proportion of abstracts (in percent) unique to each dataset for every ontology. PubMed GO has the lowest number of unique abstracts while PubMed Gene has the highest number of abstracts unique to it (except in BP).

3.2 Protein-centric Approach

For this approach, the classifiers are still trained to predict the GO terms associated with an abstract, however, the test set would consist of proteins represented by features extracted from the abstracts that have been associated with that protein in the Curated dataset. The main difference between this approach and the paper-centric approach is that here, a test instance is a feature vector extracted from the

concatenation one or more abstracts instead of a single abstract. In addition to the hierarchical F-max, we also compute the S-min scores for this approach. As with the paper-centric approach, we use 5-fold cross-validation for the Curated and Curated+PubMed Gene datasets.

We divide our testing into three categories of proteins: (a) multi-species proteins, (b) human proteins, and (c) yeast proteins.

3.2.1 Multi-species Proteins

Cellular Component (CC)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.616	0.635	0.598	0.01
PubMed GO	0.568	0.538	0.602	0.01
PubMed Gene	0.622	0.666	0.584	0.08
Curated+PubMed Gene	0.689	0.678	0.700	0.52
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	44.06	43.49	7.53	0.49
PubMed GO	43.87	42.72	9.98	0.16
PubMed Gene	45.57	44.93	7.60	0.12
Curated+PubMed Gene	46.74	46.18	7.21	0.56

Table 6: Results for multi-species proteins in CC. Curated+PubMed Gene has the best F-max while PubMed GO has the best S-min. *ru* is the remaining uncertainty and *mi* stands for misinformation. τ_1 is the decision threshold for F-max and τ_2 is the decision threshold for S-min.

In Table 6, we have the F-max and S-min scores of multi-species proteins in CC.

Curated+PubMed Gene dataset has the highest F-max followed by PubMed Gene, Curated and lastly, PubMed GO. Curated+PubMed outperforms the other datasets in both precision and recall.

Although PubMed GO has the lowest F-max, it has the best S-min. This is followed by Curated, PubMed Gene and Curated+PubMed Gene. Interestingly, Curated+PubMed Gene and PubMed GO have switched places in ranking in terms of F-max and S-min scores.

Molecular Function (MF)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.394	0.663	0.280	0.01
PubMed GO	0.408	0.533	0.331	0.01
PubMed Gene	0.426	0.549	0.348	0.01
Curated+PubMed Gene	0.566	0.620	0.521	0.42
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	92.30	91.83	9.29	0.27
PubMed GO	92.14	91.91	6.42	0.09
PubMed Gene	86.82	85.74	13.61	0.44
Curated+PubMed Gene	92.83	92.83	0.0	1.0

Table 7: Results for multi-species proteins in MF. Curated+PubMed Gene has the best F-max while PubMed Gene has the best S-min.

Table 7 shows the results for multi-species proteins in MF.

Curated+PubMed Gene has the best F-max followed by PubMed Gene then PubMed GO and Curated. PubMed GO and Curated have very close F-max scores (0.408 and 0.394, respectively). The reason for Curated's low F-max can be attributed to its poor recall.

Meanwhile, PubMed Gene has the best S-min followed by PubMed GO, Curated, and Curated+PubMed Gene although the last three have very close S-min scores (92.14, 92.30 and 92.83, respectively). The reason for PubMed Gene's good S-min score is that it has good ru. On the other hand, Curated+PubMed Gene has such poor ru that its decision threshold τ_2 has to be at 1.0 where mi is 0.

Biological Process (BP)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.385	0.610	0.282	0.01
PubMed GO	0.331	0.592	0.229	0.01
PubMed Gene	0.399	0.402	0.396	0.01
Curated+PubMed Gene	0.519	0.537	0.503	0.51
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	229.72	226.94	35.66	0.41
PubMed GO	225.88	223.81	30.50	0.04
PubMed Gene	227.63	224.03	40.32	0.12
Curated+PubMed Gene	236.00	232.89	38.21	0.98

Table 8: Results for multi-species proteins in BP. Curated+PubMed Gene has the best F-max. PubMed GO has the best S-min.

Table 8 shows the results for multi-species proteins in BP.

Curated+PubMed Gene has the best F-max by a large margin. This is followed by PubMed Gene, Curated and lastly, PubMed GO. Curated+PubMed Gene has achieved a good balance of precision and recall (0.537 and 0.503, respectively).

PubMed GO has the best S-min then PubMed Gene and Curated whose S-min scores are close to each other (227.63 and 229.72, respectively) and in last place by far is Curated+PubMed Gene.

Overall for multi-species proteins, Curated+PubMed Gene is the best dataset in terms of F-max followed by PubMed Gene. PubMed GO has the worst F-max for CC and BP while Curated has the worst F-max for MF.

In terms of S-min, PubMed GO is best for CC and BP, PubMed Gene for MF. Curated+PubMed Gene has the worst S-min for all ontologies. Curated+PubMed Gene also has the highest decision threshold (τ_2) which leads us to believe that the reason for its poor S-min is that it is very conservative in its predictions, especially for high-IC terms.

3.2.2 Human Proteins

Cellular Component (CC)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.574	0.671	0.502	0.01
PubMed GO	0.540	0.615	0.480	0.01
PubMed Gene	0.591	0.628	0.558	0.12
Curated+PubMed Gene	0.612	0.614	0.610	0.18
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	59.71	58.94	9.56	0.19
PubMed GO	60.71	60.20	7.85	0.16
PubMed Gene	56.16	54.47	13.67	0.14
Curated+PubMed Gene	57.10	53.84	19.02	0.44

Table 9: Results for human proteins in CC. Curated+PubMed Gene has the best F-max while PubMed Gene has the best S-min.

Table 9 shows the F-max and S-min scores for human proteins in CC.

Curated+PubMed Gene has the best F-max followed by PubMed Gene, Curated and lastly, PubMed GO. Curated+PubMed Gene has achieved a good balance of precision and recall (0.614 and 0.610, respectively). On the other hand, PubMed GO's low F-max can be attributed to its low recall.

PubMed Gene has the best S-min followed by Curated+PubMed Gene, Curated and lastly, PubMed GO. PubMed Gene's good S-min score is due to its good ru while PubMed GO's poor S-min is also due to it's poor ru.

Molecular Function (MF)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.372	0.526	0.288	0.01
PubMed GO	0.287	0.531	0.197	0.01
PubMed Gene	0.436	0.463	0.413	0.02
Curated+PubMed Gene	0.499	0.522	0.479	0.13
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	111.95	110.85	15.66	0.13
PubMed GO	114.16	113.79	9.21	0.08
PubMed Gene	104.37	101.70	23.44	0.15
Curated+PubMed Gene	107.60	105.62	20.54	0.50

Table 10: Results for human proteins in MF ontology. Curated+PubMed Gene has the best F-max while PubMed Gene has the best S-min.

Table 10 shows the F-max and S-min scores for human proteins in MF.

Curated+PubMed Gene has the best F-max followed by PubMed Gene, Curated, and PubMed GO. PubMed GO's poor F-max score can be attributed to its very low recall (0.197 while the next lowest recall is 0.288).

Meanwhile, PubMed Gene has the best S-min followed by Curated+PubMed Gene, Curated and lastly, PubMed GO. In general, PubMed GO is the worst-performing dataset for MF in human proteins.

Biological Process (BP)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.367	0.491	0.293	0.01
PubMed GO	0.237	0.475	0.158	0.01
PubMed Gene	0.403	0.412	0.393	0.09
Curated+PubMed Gene	0.475	0.498	0.455	0.29
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	302.00	299.40	39.57	0.20
PubMed GO	294.03	290.78	43.57	0.03
PubMed Gene	296.60	293.60	42.06	0.11
Curated+PubMed Gene	295.91	287.76	68.98	0.92

Table 11: Results for human proteins in BP ontology. Curated+PubMed Gene has the best F-max. PubMed GO has the best S-min.

Table 11 shows the F-max and S-min scores for human proteins in BP.

Curated+PubMed Gene which achieves a good balance of precision and recall (0.498 and 0.455, respectively) has the best F-max by a large margin followed by PubMed Gene, Curated and lastly, PubMed GO. The large difference between Curated+PubMed Gene and PubMed GO's F-max scores can again be attributed to PubMed GO's very low recall, 0.158 (by contrast, Curated+PubMed Gene has a recall of 0.455).

On the other hand, PubMed GO has the best S-min followed by Curated+PubMed Gene, PubMed Gene and lastly, Curated.

For human proteins, Curated+PubMed Gene has the best F-max overall. PubMed Gene has the best S-min for CC and MF while PubMed GO has the best S-min for BP. PubMed GO has the worst F-max overall because of its very low recall. PubMed GO has the worst S-min for CC and MF while Curated has the worst S-min for BP.

3.2.3 Yeast Proteins

Cellular Component (CC)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.663	0.798	0.567	0.05
PubMed GO	0.620	0.747	0.530	0.01
PubMed Gene	0.667	0.787	0.579	0.28
Curated+PubMed Gene	0.709	0.759	0.664	0.49
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	63.79	63.16	8.92	0.90
PubMed GO	56.03	55.27	9.21	0.06
PubMed Gene	51.67	49.22	15.55	0.27
Curated+PubMed Gene	68.40	68.40	0.0	1.0

Table 12: Results for yeast proteins in CC ontology. Curated+PubMed Gene has the best F-max and PubMed Gene has the best S-min.

Table 12 shows the F-max and S-min results for yeast proteins in CC.

Curated+PubMed Gene has the best F-max score followed by PubMed Gene, Curated and lastly, PubMed GO. Compared to multi-species and human proteins, all datasets here have achieved a good balance of precision and recall.

PubMed Gene has the best S-min followed by PubMed GO, Curated and lastly, Curated+PubMed Gene. Unlike in multi-species and human proteins, S-min scores here from different datasets differ widely.

Molecular Function (MF)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.470	0.555	0.407	0.01
PubMed GO	0.374	0.561	0.281	0.01
PubMed Gene	0.460	0.546	0.413	0.02
Curated+PubMed Gene	0.518	0.567	0.477	0.36
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	91.48	91.37	4.50	0.94
PubMed GO	89.30	89.16	4.98	0.08
PubMed Gene	83.41	81.42	18.10	0.44
Curated+PubMed Gene	91.67	91.67	0.0	0.99

Table 13: Results for yeast proteins in MF ontology. Curated+PubMed Gene has the best F-max and PubMed Gene has the best S-min.

Table 13 shows the F-max and S-min scores for yeast proteins in MF.

Curated+PubMed Gene again has the best F-max followed by Curated, PubMed Gene and lastly, PubMed GO. The low F-max of PubMed GO is again due to its very low recall, 0.281 (by contrast, all other datasets have recall above 0.40).

PubMed Gene has the best S-min followed by PubMed GO, then Curated and lastly, Curated+PubMed Gene although the last two have very close S-min scores (91.48 and 91.67, respectively).

Biological Process (BP)				
F-max				
Dataset	**F-max**	**Prec(τ_1)**	**Rec(τ_1)**	**Thresh τ_1**
Curated	0.442	0.558	0.365	0.01
PubMed GO	0.295	0.637	0.192	0.01
PubMed Gene	0.443	0.528	0.383	0.27
Curated+PubMed Gene	0.525	0.570	0.487	0.70
S-min				
Dataset	**S-min**	**ru(τ_2)**	**mi(τ_2)**	**Thresh τ_2**
Curated	218.46	218.46	0.0	1.0
PubMed GO	206.25	201.63	43.41	0.02
PubMed Gene	197.92	194.13	38.53	0.30
Curated+PubMed Gene	218.46	218.46	0.0	1.0

Table 14: Results for yeast proteins in BP ontology. Curated+PubMed Gene has the best F-max. PubMed Gene has the best S-min.

Table 14 shows the F-max and S-min results for yeast proteins in BP.

Curated+PubMed Gene has the best F-max followed by PubMed Gene and Curated (their F-max scores differ by only 0.001) and lastly, PubMed GO. Although none of the datasets managed a good balance of precision and recall, the worst imbalance is displayed by PubMed GO with a precision of 0.637 and recall of 0.192. In general, PubMed GO has very low recall in BP.

PubMed Gene has the best S-min followed by PubMed GO and then Curated and Curated+PubMed Gene, both with the same S-min at the same decision threshold. In both cases, *mi* decreases rapidly as the threshold increases while *mi* only decreases slowly therefore the highest threshold (1.0) achieves the minimum semantic distance.

Overall, Curated+PubMed Gene has the best F-max for yeast proteins while PubMed GO has the lowest F-max scores. PubMed Gene has the best S-min for yeast proteins followed by PubMed GO then Curated and Curated+PubMed Gene. Curated and Curated+PubMed Gene tend to have close S-min scores.

For all three protein categories, Curated+PubMed Gene has the best F-max score. The overall ranking of datasets in terms of F-max is as follows:

1. Curated+PubMed Gene
2. PubMed Gene

3. Curated
4. PubMed GO

Except for BP, this ranking follows the same ranking for number of abstracts in each dataset (see Table 3). In BP, PubMed GO has more abstracts than Curated.

In terms of S-min, however, there is no definite ranking of datasets. For multi-species proteins, PubMed GO has the best S-min for CC and BP, PubMed Gene for MF. The worst is Curated+PubMed Gene for all ontologies. For human proteins, PubMed Gene has the best S-min for CC and MF and PubMed GO for BP. PubMed GO has the worst S-min for CC and MF while Curated has the worst S-min for BP. For yeast, PubMed Gene has the best S-min while Curated+PubMed Gene has the worst.

This is a rather surprising result since Curated+PubMed Gene has the best F-max overall and PubMed GO has the worst F-max but in terms of S-min, we are seeing the opposite in multi-species and yeast proteins: PubMed GO has the best results while Curated+PubMed Gene has the worst. In general, for all datasets it is ru that contributes most to the S-min score while mi contributes very little. None of the datasets managed to achieve a good balance between remaining uncertainty and misinformation. This tells us that at the decision threshold τ_2, our classifiers have the tendency to under-predict especially for high-IC terms which leads to more false negatives (high ru).

On the question of which of our datasets is best for protein function prediction, the answer is that it depends on the evaluation metric, the ontology (CC, MF or BP) and the proteins (human, yeast or multi-species). If we are concerned only with F-max and treat all GO terms equally then Curated+PubMed Gene is the best. If we are concerned with the IC's of the terms we are predicting then PubMed Gene is best except for multi-species proteins in CC where PubMed GO is best.

Another observation is that in general, performance is better in protein-centric prediction than paper-centric prediction. This could be because in protein-centric predictions, we are concatenating one or more abstracts which provides a more comprehensive view of a protein as opposed to just a single abstract which is specific to a particular protein and a particular annotation. From the inter-related nature of the GO, we know that an annotation can imply another annotation. For instance, if a protein has two or more annotations that have a common ancestor, we are more likely to also predict these commonalities. Concatenating abstracts helps the classifier infer these related annotations. We speculate that this is also the reason

Curated+PubMed Gene has the best F-max in the protein-centric approach but the worst in the paper-centric approach.

Moreover, we evaluate precision and recall per test instance. In paper-centric predictions this means that one test instance may have only one term associated with it (before propagation) and the classifier has to get the one term correct or at least something close to it. Compare this to the situation in protein-centric predictions where we are more likely to have multiple terms associated with a single test instance and getting one term wrong will have less of an impact on the F-max as it would in the paper-centric case.

We investigate this further by counting the number of terms associated with papers and proteins. For CC, the mean number of (non-propagated) terms associated with a paper is 1.59, and with proteins, 2.01. For MF, the mean is 1.59 for papers and 1.82 for proteins. For BP, the mean is 2.0 for papers and 3.29 for proteins. Therefore in all these ontologies, proteins on average have more annotations than papers. These numbers do not include the terms' ancestors because once we have predicted a term, propagation is trivial.

The histograms in Figures 19 to 21 show the number of terms (annotations) associated with papers and proteins.

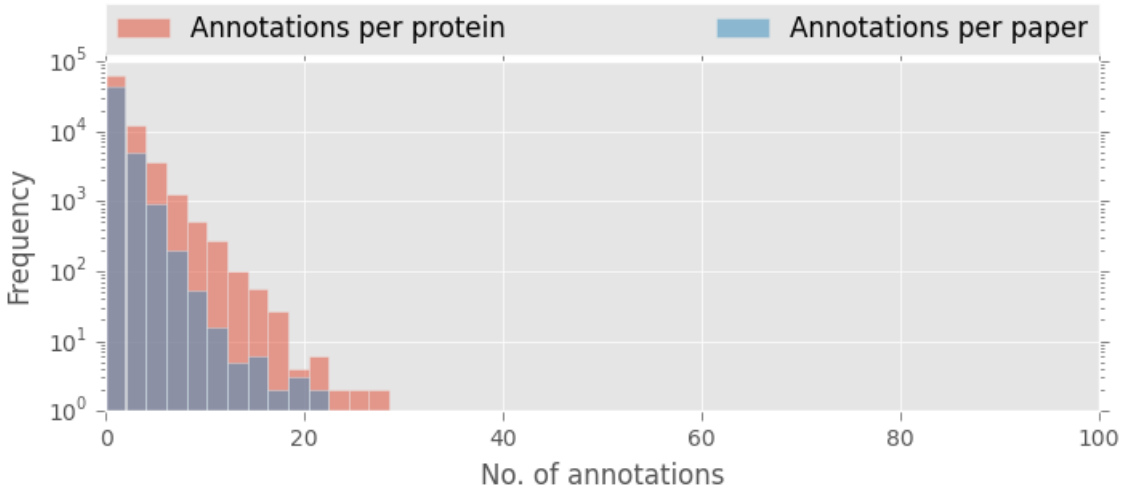

Figure 19: Number of annotations per paper and per protein in CC.

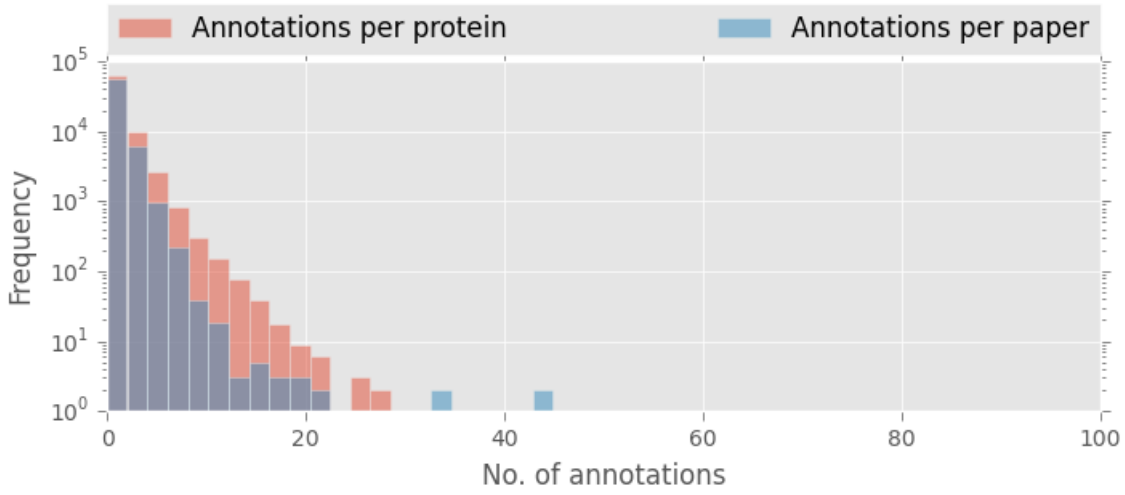

Figure 20: Number of annotations per paper and per protein in MF.

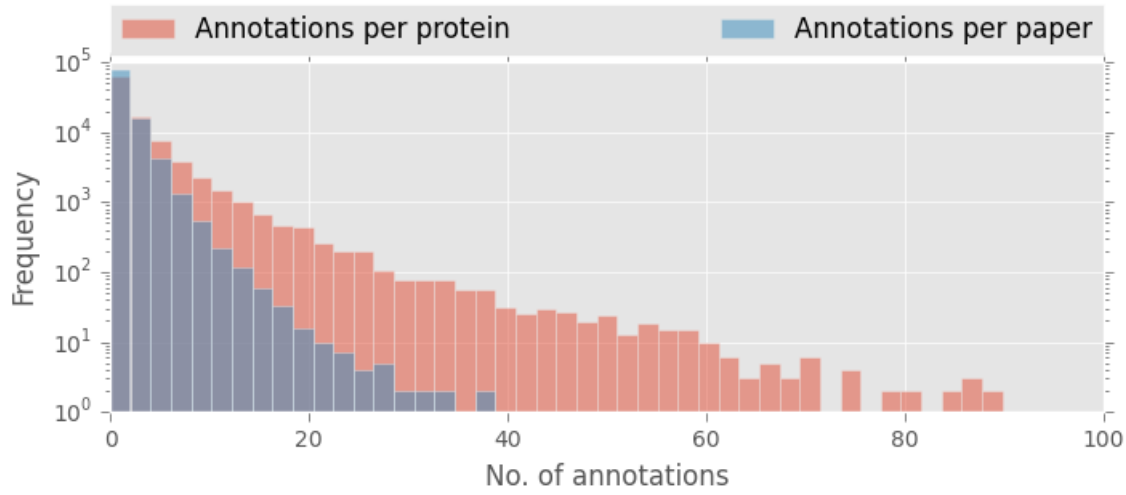

Figure 21: Number of annotations per paper and per protein in BP.

In protein categories, yeast proteins has better F-max than human proteins for all datasets. However human proteins have more annotations per protein than yeast proteins therefore the number of annotations alone does not explain this behaviour (see Figure 22). But when we look at the IC of the terms associated with these proteins, we find that human proteins are more likely to be annotated with high-IC terms than yeast (see Figure 23). This also explains why human proteins have higher F-max than multi-species proteins since Figure 23 also shows that there are more human proteins associated with terms that have IC below 13 than multi-species

proteins. In the results for the term-centric approach in the next section, we will see that there is a negative correlation between F-max and IC.

In terms of S-min, the situation is not as straightforward. In CC, multi-species proteins have the best S-min followed by yeast then human. For MF and BP, yeast has the best S-min and human has the worst. This means all ontologies perform worst on human proteins when we take into account the IC's of incorrectly predicted terms even though there are more human proteins than multi-species proteins associated with terms with IC below 13. We speculate that the reason for this is that there are some terms associated only with multi-species proteins that have high IC but whose names express general concepts that are easy to find in literature. An example is GO:0018995 (host) with an IC of 8.80. We discuss this further in Section 3.3.1, in the term-centric approach for multi-species proteins.

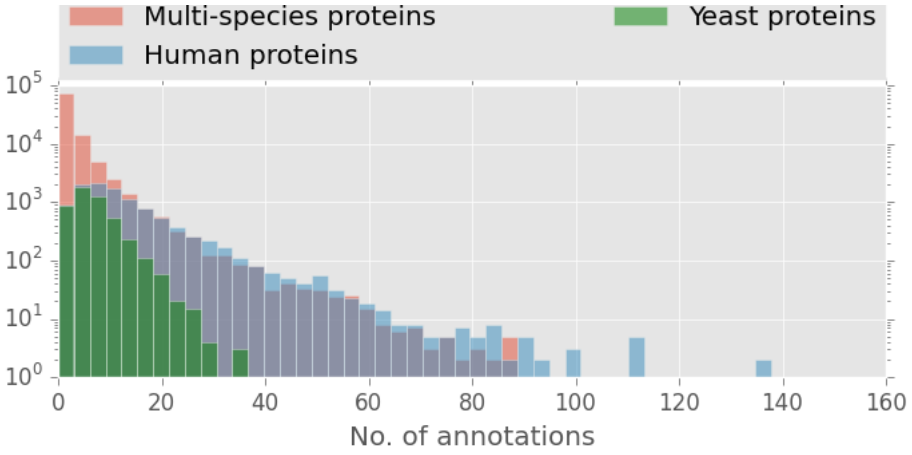

Figure 22: Number of annotations per protein.

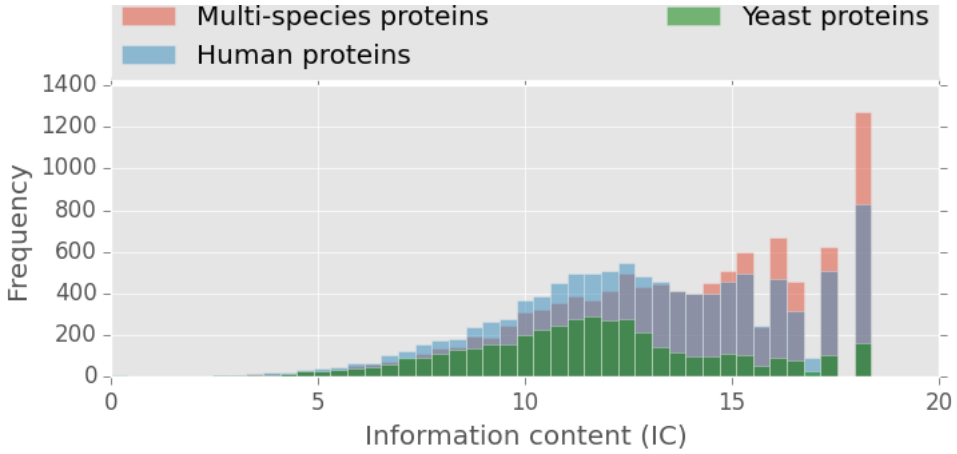

Figure 23: Histogram showing the information content (IC) of protein annotations for multi-species, human and yeast proteins.

3.3 Term-Centric Approach

The purpose of the term-centric approach is to find out which GO terms (excluding the root terms) are easy to assign and investigate some characteristics of these terms to understand their behaviour. As with the protein-centric approach, we also divide our testing into multi-species, human and yeast proteins.

3.3.1 Multi-species Proteins

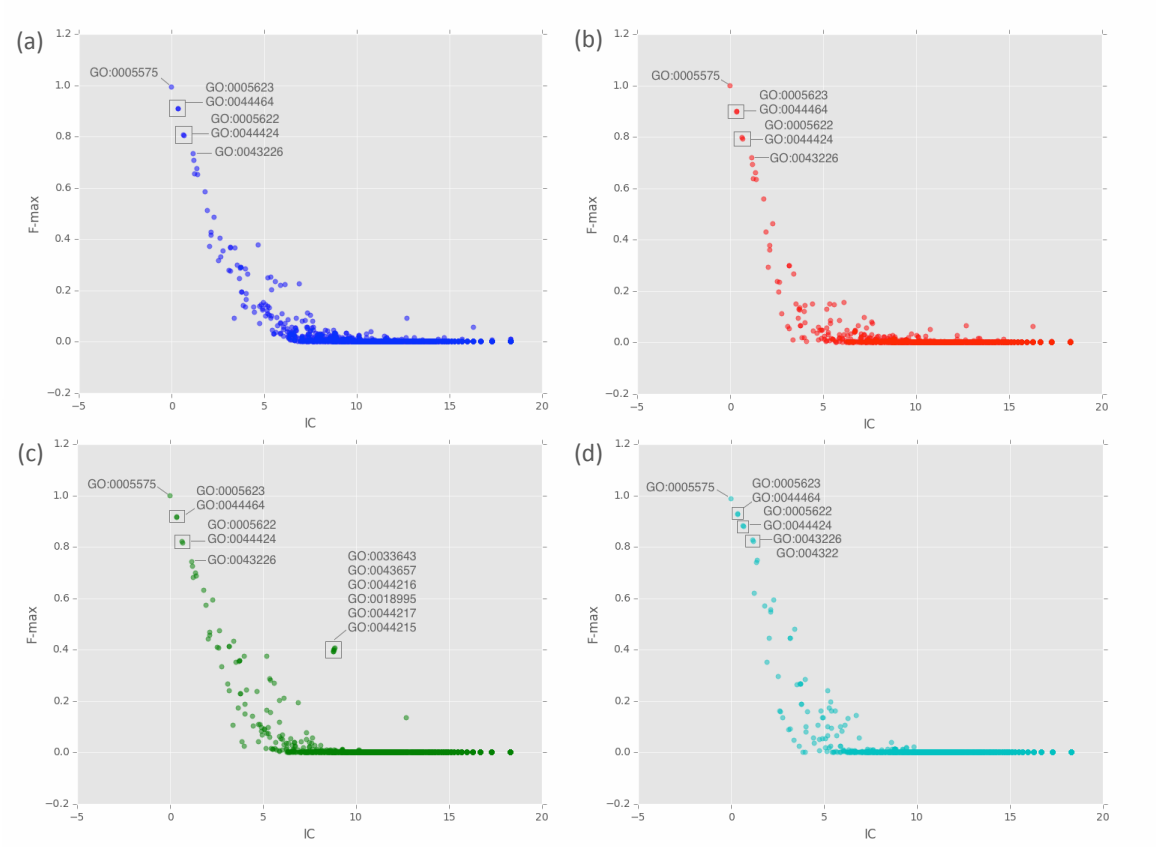

Figure 24: Relationship between performance (F-max) in multi-species proteins and information content (IC) of GO terms in CC for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In Figure 24, we plotted the relationship between IC and F-max of GO terms in cellular component (CC) ontology for multi-species proteins. We observe that we have the same set of top five highest-performing GO terms (excluding the root) in the same order for all datasets:

1. GO:0005623 (cell), $IC = 0.35$
2. GO:0044464 (cell part), $IC = 0.35$
3. GO:0005622 (intracellular), $IC = 0.63$
4. GO:0044424 (intracellular part), $IC = 0.68$
5. GO:0043226 (organelle), $IC = 1.18$

The term with the highest IC in the set is GO:0043226 with an IC of 1.18 while the lowest IC is shared by GO:0005623 and GO:0044464 both with an IC of 0.35. Three

of the terms are direct children of the root while the other two are two edges away from the root. Of these five terms, only two are not present in the gold standard annotations (GO:0044464 and GO:0044424). According to UniProt-GOA, these two terms are not to be used for direct annotations, meaning that we should not assign these terms as the most specific annotation of a protein [BDH+09].

In Figure 24c, we have six GO terms that have high F-max scores relative to their IC's. These terms are:

1. GO:0033643 (host cell part), $IC = 8.87$
2. GO:0043657 (host cell), $IC = 8.84$
3. GO:0044216 (other organism cell), $IC = 8.82$
4. GO:0018995 (host), $IC = 8.80$
5. GO:0044217 (other organism part), $IC = 8.79$
6. GO:0044215 (other organism), $IC = 8.78$

All these terms are related to GO:0044215 (other organism), since GO:0018995 (host) is a child of GO:0044215. The term "host organism" refers to any organism from which another organism such as a parasite gains nourishment. Other organism refers to any organism with which the organism being studied interacts with [BDH+09]. We speculate that the reason for these terms' relatively high performance given their IC is that they express generic concepts that are easy to find in literature despite their low presence in the GOA annotations. Of these six terms, GO:0033643 and GO:0044217 are not to be used as direct annotations.

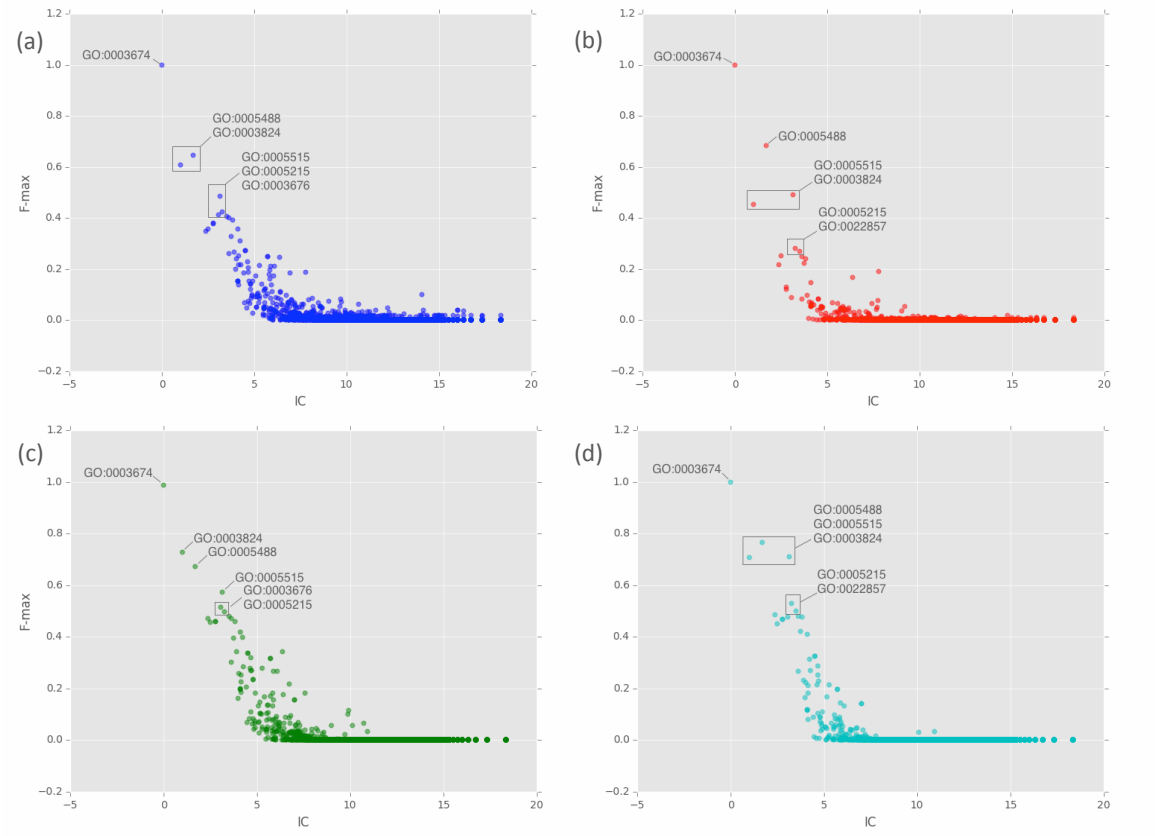

Figure 25: Relationship between F-max in multi-species proteins and IC of GO terms in MF for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In Figure 25, we have the relationship between F-max and IC for molecular function (MF) terms. GO:0005488 (binding) and GO:0003824 (catalytic activity) are consistently the top two best-performing terms. Both are direct children of the root. GO:0005515 (protein binding), a child of 0005488, is also consistently in the top five which is not surprising since this term originally made up almost one-third of the MF annotations in our dataset. Even though we have subsequently removed all annotations of GO:0005515 from our dataset, it can still be in predicted label set because when any of its children are used as annotations, GO:0005515 is also understood to be an annotation (true path rule). These terms aside, the datasets are not as consistent as in CC on which terms are the easiest to assign. Only GO:0005488 is not to be used for direct manual annotation.

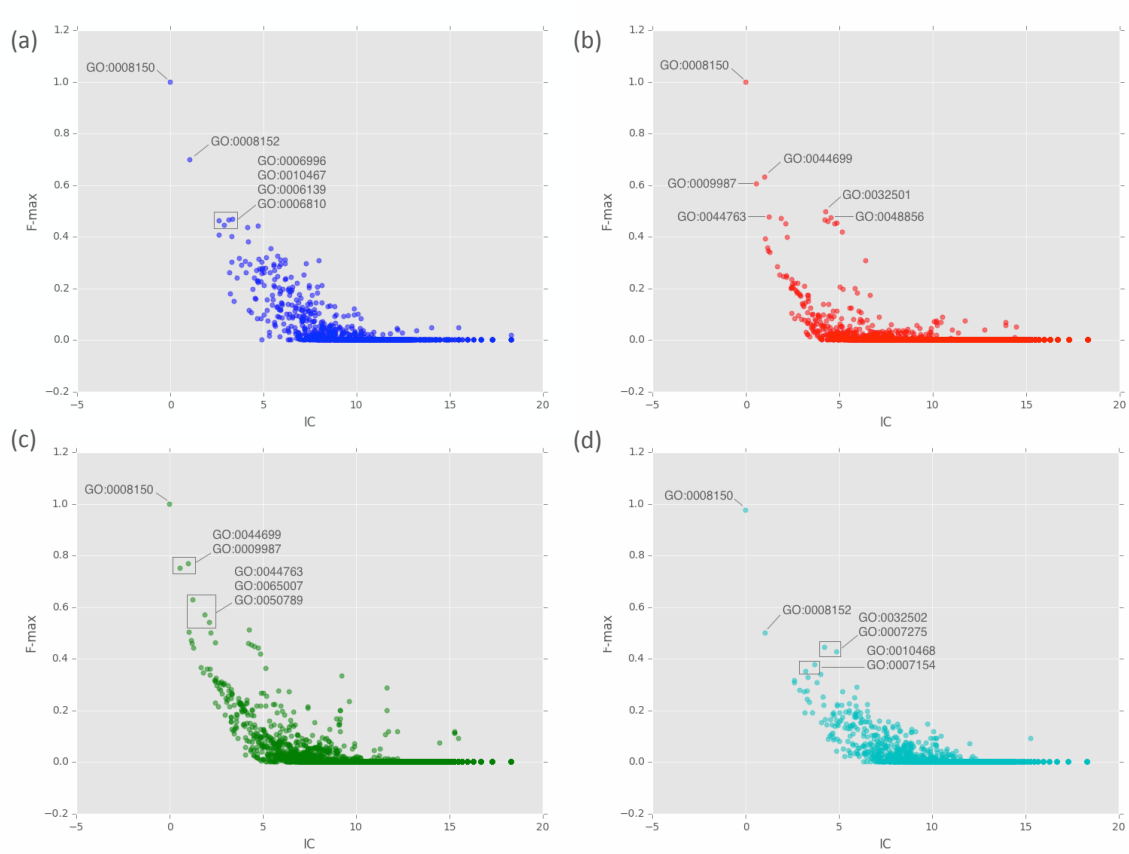

Figure 26: Relationship between F-max in multi-species proteins and IC of GO terms in BP for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

Figure 26 shows the F-max and IC of biological process (BP) terms for multi-species proteins. We observe that there is even less consistency across datasets on the top-scoring terms. For instance, GO:0009987 (cellular process) and GO:0044699 (single-organism process), both child terms of the root, are top-scoring terms for all datasets except in Curated and PubMed Gene+Curated and in those datasets, four of their top-scoring terms do not appear in the top five of the other datasets. Of the top terms, only GO:0008152 (metabolic process), another child of the root, should not be used for direct manual annotation [BDH+09].

3.3.2 Human Proteins

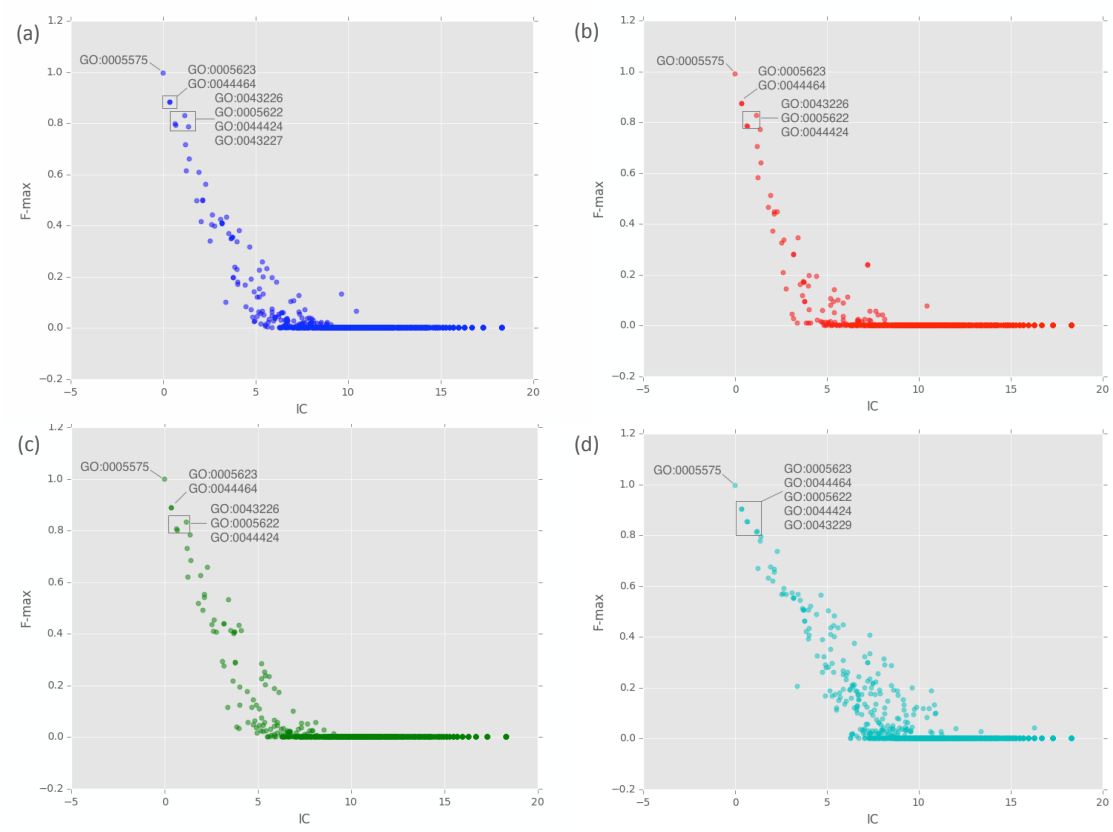

Figure 27: Relationship between F-max in human proteins and IC of GO terms in CC for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

Figure 27 shows the relationship between F-max and IC of CC terms for human proteins. We have the same top five best-scoring terms for all datasets except for Curated+PubMed Gene. This dataset does not have GO:0043226 (organelle) in its top five instead it has GO:0043229 (intracellular organelle), a child term of GO:0043226. However it does have GO:0043226 in sixth place and the difference of their F-max scores is negligible (0.814 and 0.811). Other than this slight difference, the top CC terms are still consistent across datasets.

Unlike in multi-species proteins, we don't see the same behaviour in PubMed Gene regarding terms related to hosts and other organisms. This is because in our dataset, there are no human proteins associated with those terms.

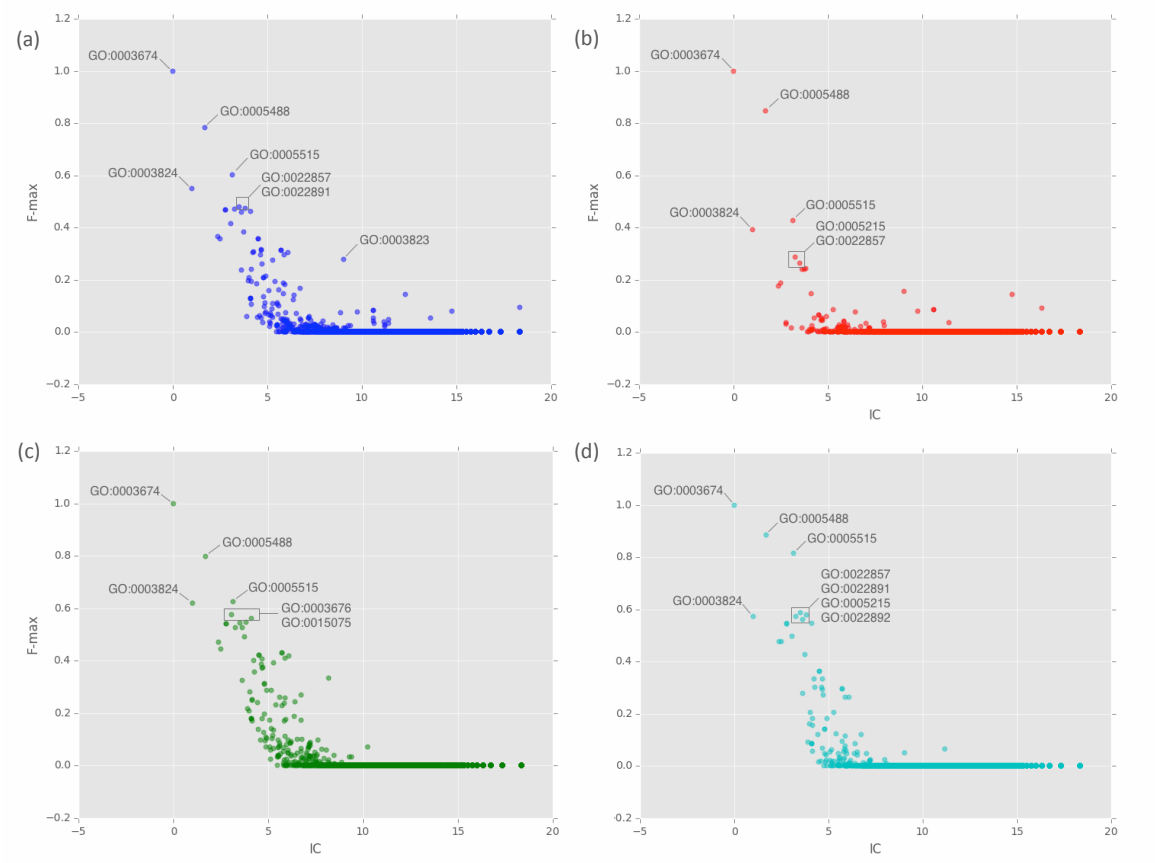

Figure 28: Relationship between F-max in human proteins and IC of GO terms in MF for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In Figure 28, we have the F-max plotted against IC of MF terms for human proteins. As with the multi-species proteins, we observe that the three terms that appear in the top five for all datasets are GO:0005488 (binding), GO:0005515 (protein binding) and GO:0003824 (catalytic activity). Of these three, the term with the highest IC is GO:0005515 with IC of 3.16 while GO:0003824 has the lowest IC, 1.02.

In Curated, PubMed GO and Curated+PubMed Gene, we have three related terms whose performance are close to each other: GO:0005215 (transporter activity), GO:0022857 (transmembrane transporter activity) and GO:0022891 (substrate-specific transmembrane transporter activity). GO:0005215 is parent of GO:0022857 which is a parent of GO:0022891. These three terms also have "part-of" relations with BP terms.

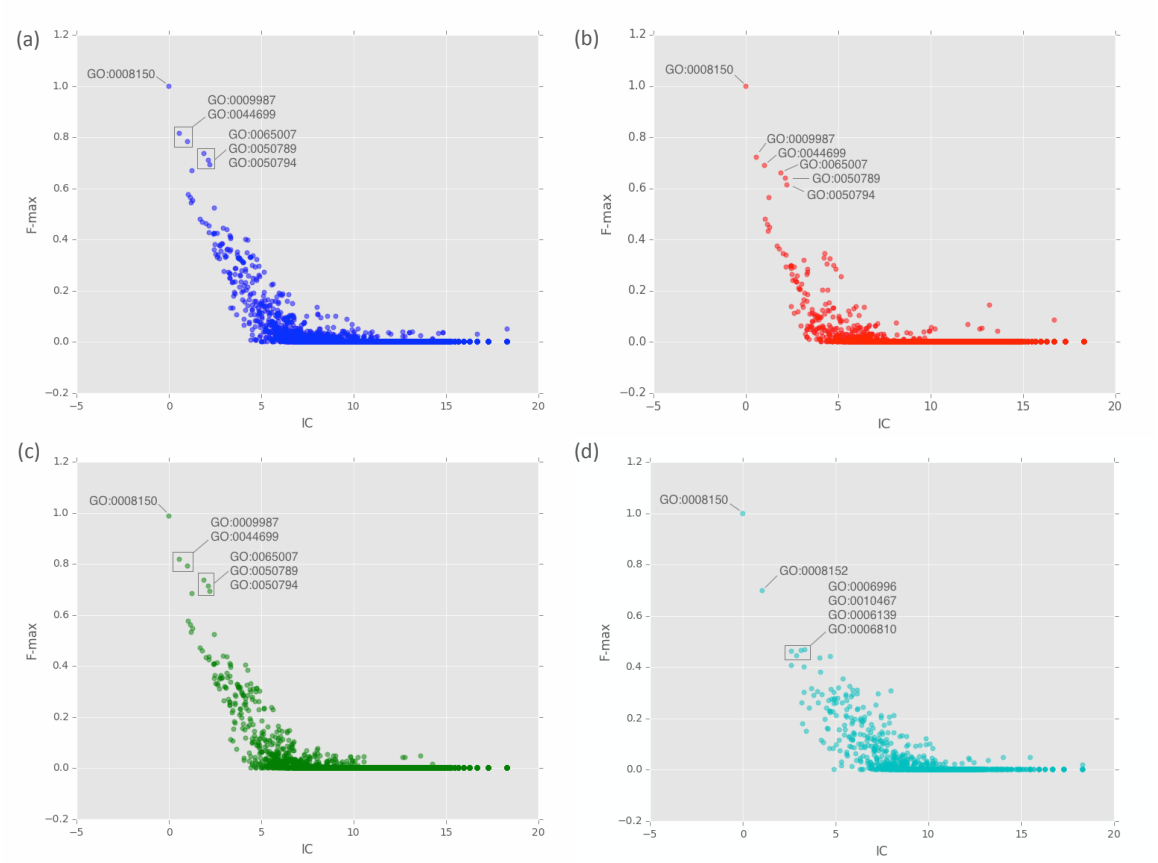

Figure 29: Relationship between F-max in human proteins and IC of GO terms in BP for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In Figure 29 we see more consistency in the top BP terms for human proteins than we did for multi-species proteins. For instance, all datasets except Curated+PubMed Gene have the same top five terms in the same order. Of these terms, GO:0050794 has the highest IC, 2.21. Interestingly, none of top five terms in Curated+PubMed Gene appear on the top five of the other datasets.

3.3.3 Yeast Proteins

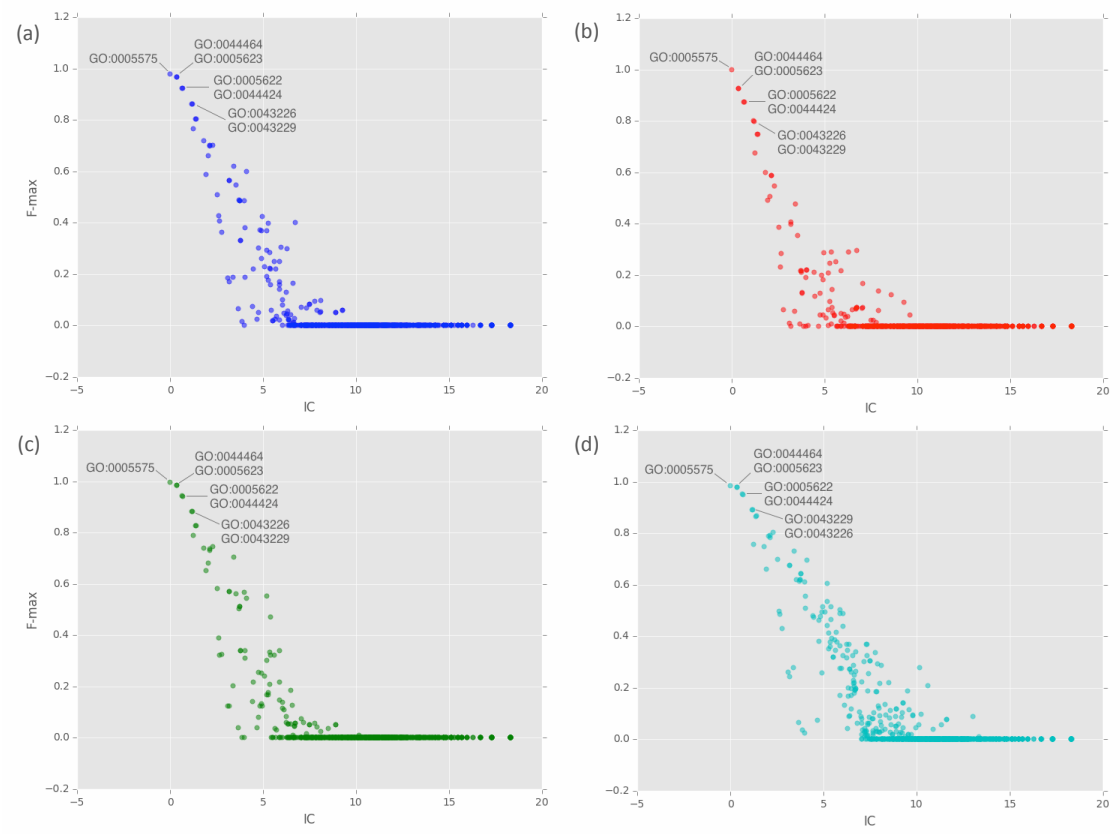

Figure 30: Relationship between F-max in yeast proteins and IC of GO terms in CC for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In Figure 30, we observe that the top six terms in CC for yeast proteins resembles that of multi-species proteins, specifically that the all datasets have the same terms in almost the same order. We include six terms here instead of five because both GO:0043226 (organelle) and GO:0043229 (intracellular organelle), have almost the same F-max in all four datasets which is not surprising since GO:0043229 is a child of GO:0043226.

In general, these graphs resemble the ones for multi-species and human proteins in that as we start from the root and move lower-right corner, we see the number of points slowly increasing which indicates that there is no sudden decrease in F-max scores as the IC increases.

Overall, for CC we observe that the terms with the highest F-max scores remain

consistent. The top CC term for all datasets for multi-species and human proteins is GO:0005623 (cell), for yeast it is GO:0044464 (cell part). Both terms are direct children of the root but GO:0044464 is also a child term of GO:0005623. For all proteins, Curated+PubMed Gene is the best dataset however PubMed Gene is not far behind. Moreover, PubMed Gene performs well on terms related to hosts and other organisms while the others perform poorly on these terms.

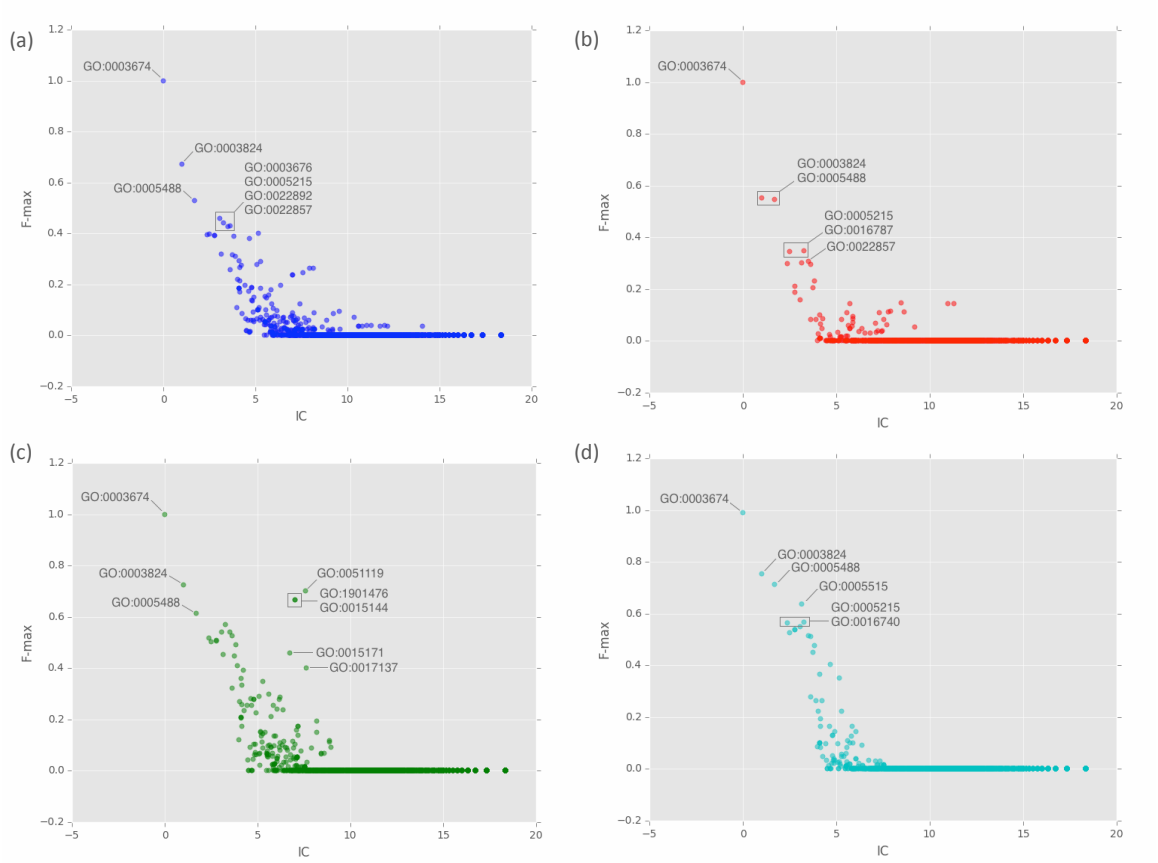

Figure 31: Relationship between F-max in yeast proteins and IC of GO terms in MF for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

For MF terms in yeast proteins (Figure 31), GO:0003824 (catalytic activity) and GO:0005488 (binding) appear in the top five for all datasets. Aside from these two terms, there is little consistency in the top terms across the four datasets compared to human and multi-species proteins.

We observe some interesting behaviour for PubMed Gene with regards to five terms with relatively high F-max scores given their IC:

1. GO:0051119 (sugar transmembrane transporter activity), $IC = 7.57$

2. GO:1901476 (carbohydrate transporter activity), $IC = 7.01$
3. GO:0015144 (carbohydrate transmembrane transporter activity), $IC = 7.01$
4. GO:0015171 (amino acid transmembrane transporter activity), $IC = 6.74$
5. GO:0017137 (Rab GTPase binding), $IC = 7.62$

As might be apparent from their names, the first four terms are all descendants of GO:0005215 (transporter activity). The first three are closely related (GO:0015144 is a parent to both GO:0051119 and GO:1901476) while the fourth term, GO:0015171, is farther away. Unlike the host organism terms, these terms express non-generic concepts. These four terms also have ancestors with "is-a" and "part-of" relations with BP terms. The fifth term, which has the highest IC of the five, is unrelated to the other four and doesn't have any relations to other ontologies. These observations aside, we do not have a plausible explanation for the F-max scores of these terms.

Overall, there is less consistency in the top terms across datasets and groups of proteins for MF as compared to CC. We also observe that the top terms in MF have higher IC's in general than in CC. For MF, the top term is either GO:0005488 (binding) or GO:0003824 (catalytic activity), both child terms of the root. Another observation is that there are some terms that are in the top five only for PubMed Gene dataset and they usually have BP ancestors. It is interesting to note that even though PubMedGene+Curated dataset is merely a union of two existing datasets, it still cannot predict some terms as well as PubMed Gene.

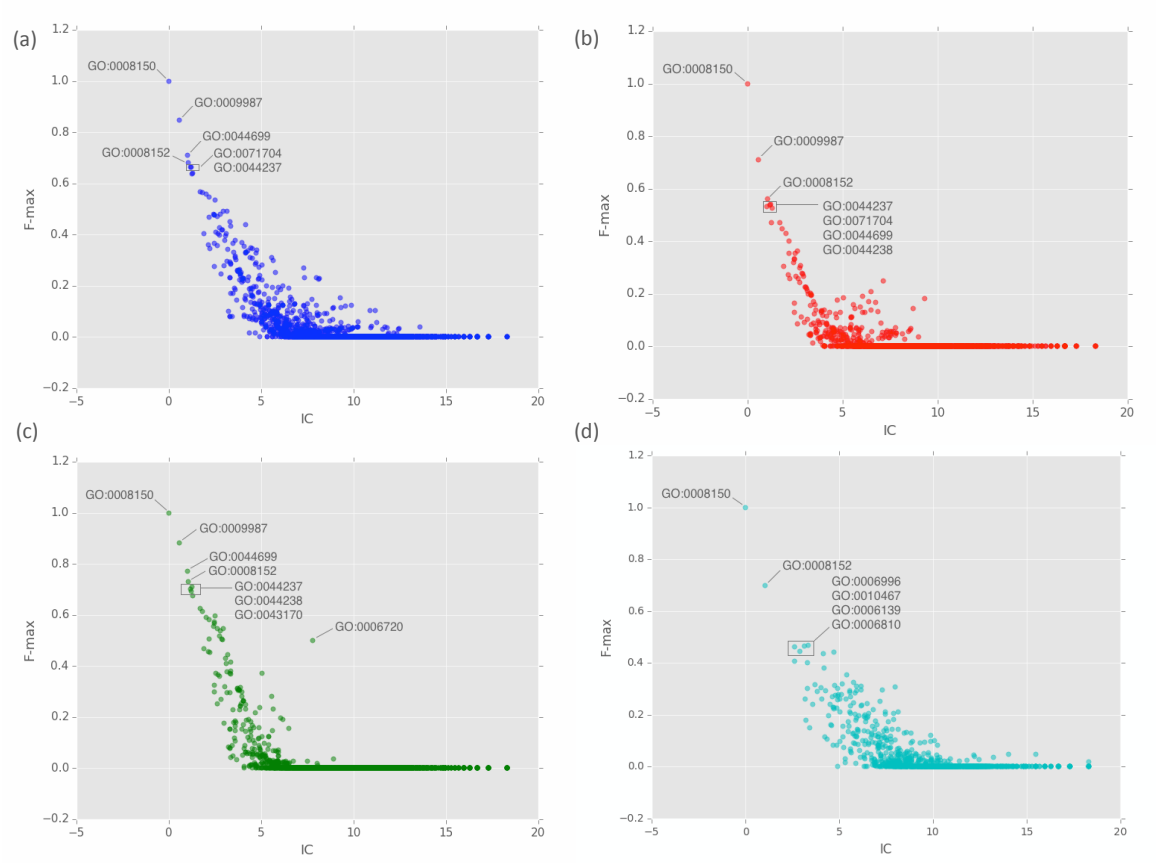

Figure 32: Relationship between performance (F-max) in yeast proteins and information content (IC) of GO terms in BP for the four datasets: (a) Curated, (b) PubMed GO, (c) PubMed Gene, and (d) Curated+PubMed Gene.

In yeast proteins for BP (Figure 32), we see little consistency in top terms across datasets. The only term present in all datasets is GO:0008152 (metabolic process), a child of the root with IC of 1.05. Curated and PubMed GO datasets have the same top five terms.

We again see an interesting behaviour in PubMed Gene, namely GO:0006720 (isoprenoid metabolic process). This term has IC= 7.79 and F-max= 0.5 and is located at level 4 of the BP ontology. Surprisingly, this term does not appear in the gold standard annotations for yeast meaning that it is an ancestor of terms that are used direct annotations but is not itself used as a direct annotation.

For BP, PubMed Gene is better for high-IC terms than Curated+PubMed Gene. This is in contrast to human and multi-species proteins where Curated+PubMed Gene tends to have higher-IC terms in its top five. Compared to CC and MF, BP is the least consistent in its top terms. For multi-species and human proteins, the top

term is either GO:0009987 (cellular process) or GO:0044699 (single-organism process), both child terms of the root. However in yeast proteins, it is either GO:0009987 or GO:0008152 (metabolic process), another child of the root.

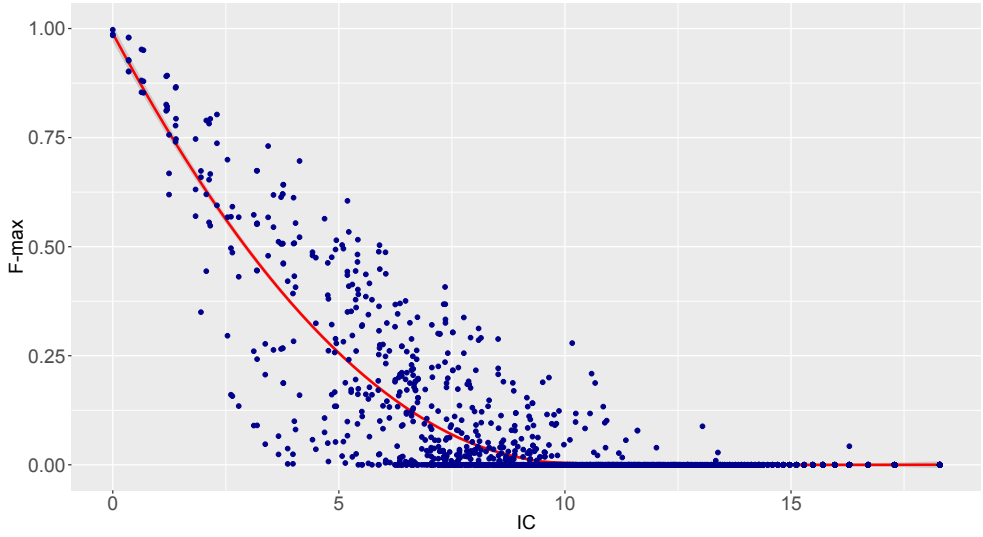

Figure 33: Relationship between F-max from Curated+PubMed Gene dataset and IC of CC terms for all proteins

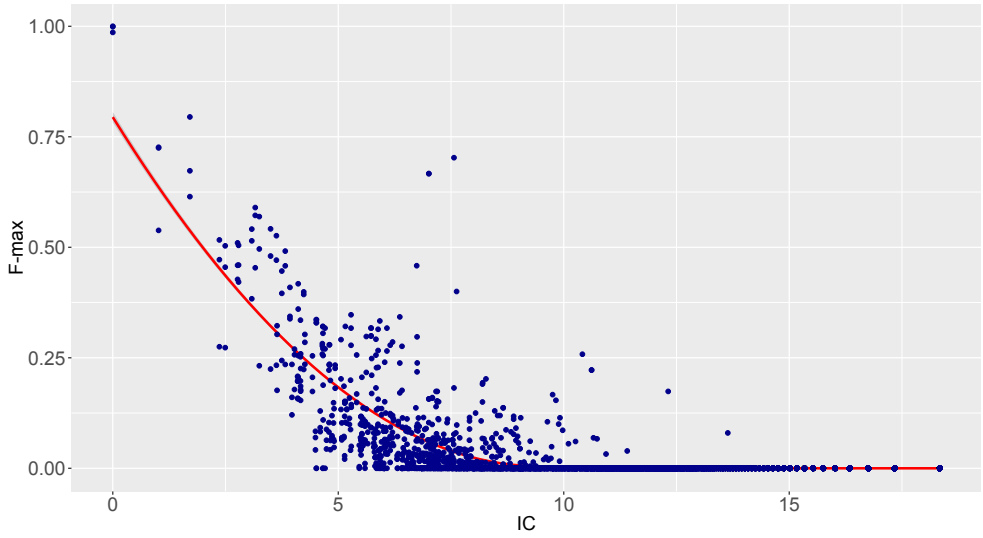

Figure 34: Relationship between F-max from PubMed Gene dataset and IC of MF terms for all proteins

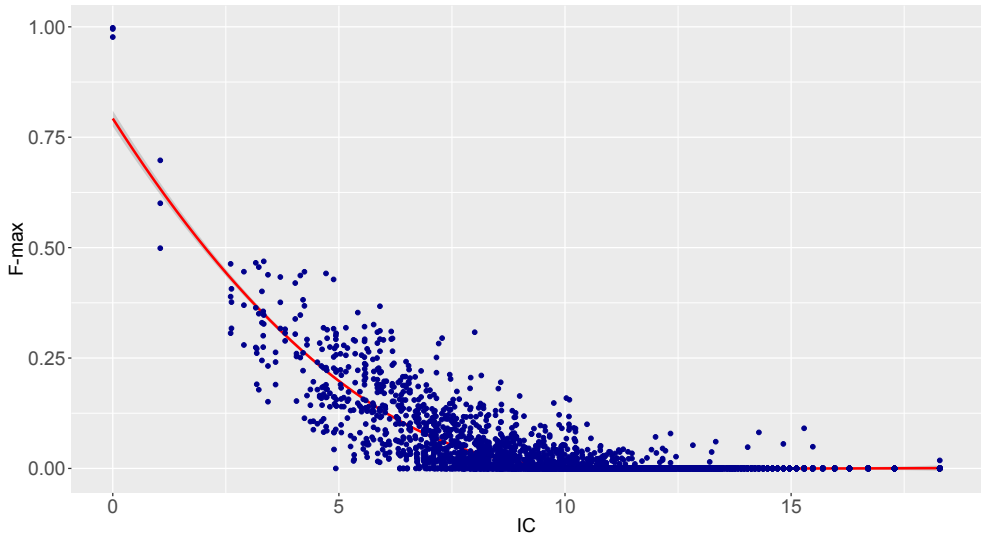

Figure 35: Relationship between F-max from Curated+PubMed Gene dataset and IC of BP terms for all proteins

As we have observed from the previous figures, there is a negative correlation between F-max and information content (IC) where terms with lower IC tend to have higher F-max scores while terms with higher IC tend to have lower F-max scores. This is expected since the larger GO classes would have more proteins annotated with it which means more data for the classifiers to train on. Moreover, the larger GO classes tend to be the more generic ones with names that are more likely to be found in literature, although there are exceptions as we have seen with GO:0018995 (host). To quantify this correlation, we compute Pearson's correlation coefficient ($-1 \geq r \leq +1$) between the IC's and F-max scores of the terms. For CC, this negative correlation is strongest in the Curated+PubMed Gene dataset ($r = -0.567$), for MF the correlation is strongest using the PubMed Gene dataset ($r = -0.405$) and for BP, it is strongest using Curated+PubMed Gene ($r = -0.556$). Correlation coefficients are computed using all proteins in the dataset (see Figures 33 to 35).

Aside from IC, we also want to investigate whether there is any relationship between a node's centrality measure and its performance (F-max). Compared to IC, there is a weaker correlation between closeness centrality and F-max. For CC and MF, the positive correlation between F-max and closeness centrality is strongest with the F-max scores from the Curated dataset ($r = 0.277$ and $r = 0.301$, respectively), for BP, the F-max scores from Curated+PubMed Gene has the strongest correlation ($r = 0.383$) (see Figures 36 to 38).

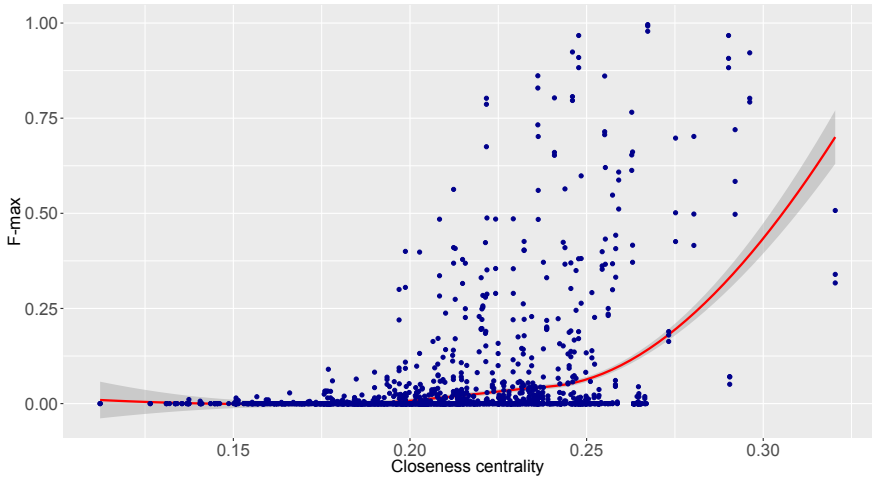

Figure 36: Relationship between F-max scores from the Curated dataset and closeness centrality of CC terms.

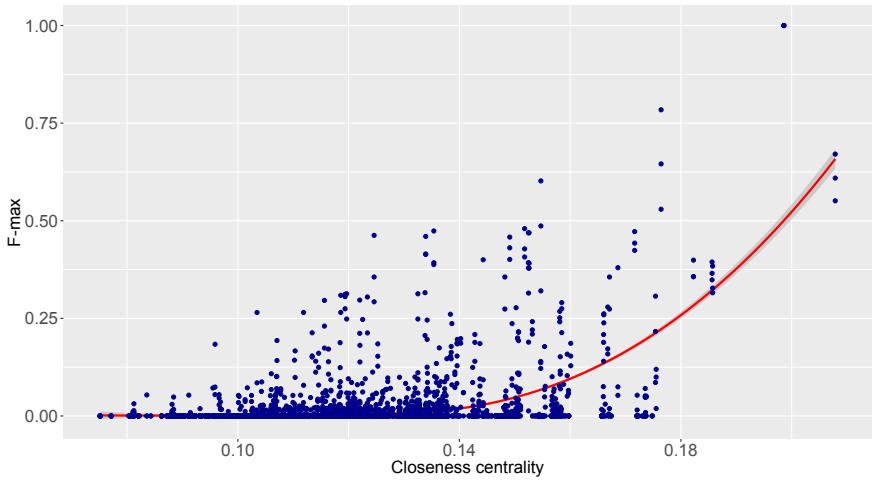

Figure 37: Relationship between F-max scores from the Curated dataset and closeness centrality of MF terms.

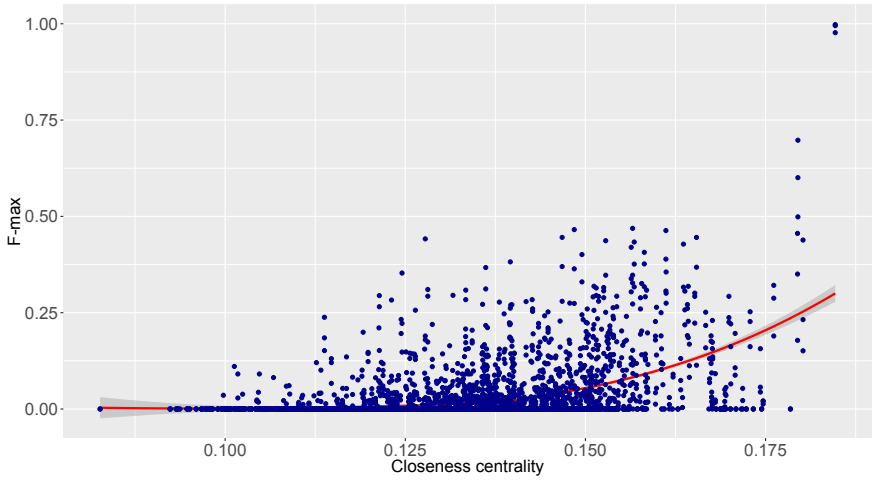

Figure 38: Relationship between F-max scores from Curated+PubMedGene and closeness centrality of BP terms.

Surprisingly, there is a stronger correlation between the betweenness centrality and the F-max of a term compared to IC and closeness centrality. In Figures 39 to 41, we see this correlation for most of the data points. For CC, this is strongest with the F-max scores from PubMed Gene ($r = 0.436$), for MF, it is strongest with PubMed GO ($r = 0.758$) and for BP, Curated+PubMed Gene ($r = 0.531$).

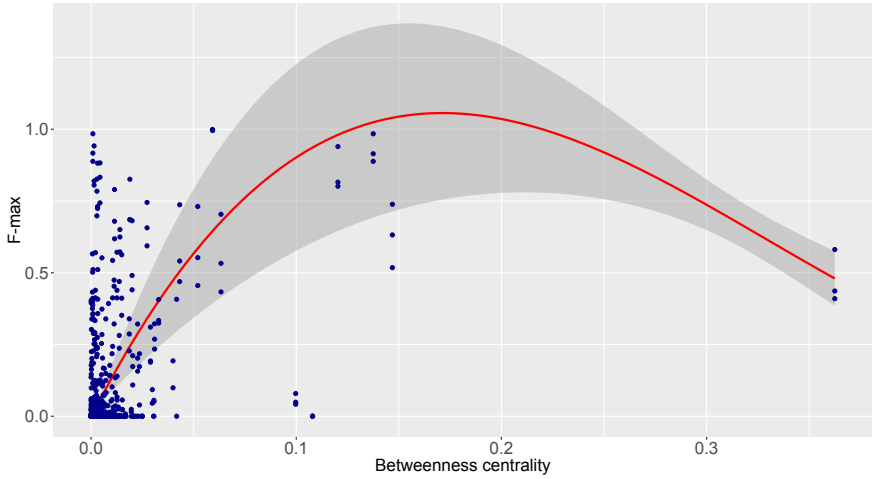

Figure 39: Relationship between F-max scores from PubMed Gene and betweenness centrality of CC terms.

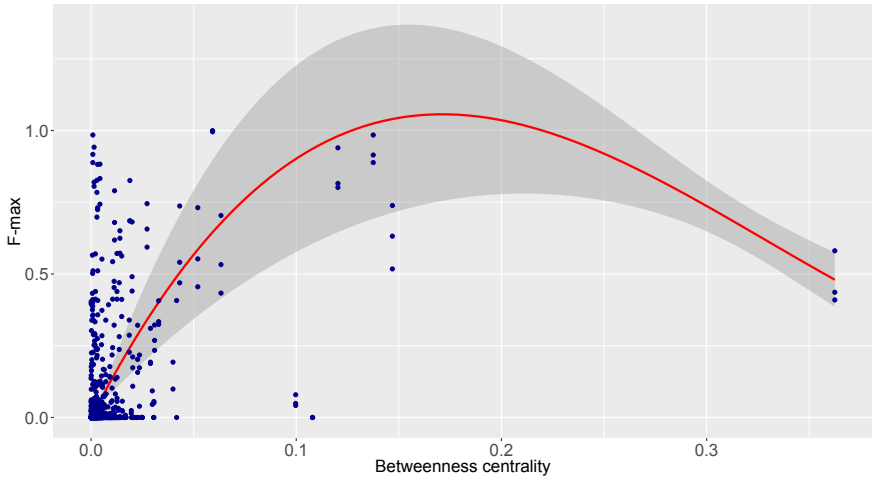

Figure 40: Relationship between F-max from PubMed GO and betweenness centrality of MF terms.

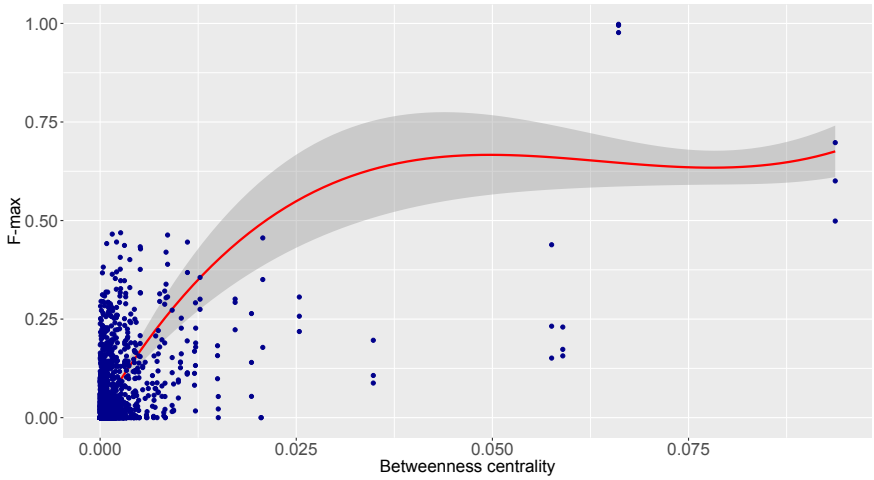

Figure 41: Relationship between F-max Curated+PubMed Gene and betweenness centrality of BP terms.

4 Comparison with the state-of-the-art

We compare the performance of our classifier with GOstruct [FKBHV15] because it is currently the state-of-the-art literature-based protein function predictor. As much as possible, we try to make a fair comparison by using the same proteins and the same number of GO classes as GOstruct.

GOstruct only included human and yeast proteins in their training and testing. Moreover, they only included GO classes that have been associated with at least 10

proteins (either human or yeast) and only annotations with experimental evidence codes (EXP, IDA, IPI, IMP, IGI, IEP, TAS). They do not specify the exact version of the GOA gold standard annotations however they mention that their test set includes 13,400 human proteins and 4,500 yeast proteins. Since their method was published in 2015 and their literature data was collected on October 2013, we checked the human and yeast annotations in the GOA archive uploaded between October 2013 and October 2014 to find the one that fit closest to this description. This is how we came to use the June 2014 annotations. We then use this as the gold standard annotations for our classifier. For this comparison, we only use the Curated+PubMed Gene dataset since it has the best F-max. Table 15 shows the results of this comparison.

Human Proteins				
	Our Classifier	**No. of classes**	**GOstruct**	**No. of classes**
CC	0.626	354	0.608	345
MF	0.542	415	0.408	509
BP	0.482	1251	0.461	2088
Yeast Proteins				
	Our Classifier	**No. of classes**	**GOstruct**	**No. of classes**
CC	0.722	354	0.719	345
MF	0.574	415	0.629	509
BP	0.519	1251	0.634	2088

Table 15: Comparison of GOstruct and our classifier using the June 2014 GO annotations.

Since we only create a classifier if our training data has at least 5 positive instances and we have less data overall than GOstruct, we have less classes than GOstruct for molecular function (MF) and biological process (BP) ontologies. Overall, our classifier performs better for human proteins and worse for yeast proteins except for cellular component (CC) where the F-max scores are very close (0.722 and 0.719).

For human proteins in CC, our F-max is a little higher despite the fact that we have 9 more classes then them. MF is surprisingly the worst-performing ontology for GOstruct. This is where we see the largest gap in performance between our method and theirs with difference of almost 14%. There is no clear winner however for BP since our method is better by 2% but they are dealing with 837 more classes.

Our performance for yeast proteins is better than human proteins however it is not as significant as the increase shown by GOstruct. For CC, there is little difference

Human Proteins				
	Our Classifier	**No. of classes**	**GOstruct**	**No. of classes**
CC	0.617	511	0.608	345
MF	0.526	707	0.408	509
BP	0.480	2026	0.461	2088
Yeast Proteins				
	Our Classifier	**No. of classes**	**GOstruct**	**No. of classes**
CC	0.719	511	0.719	345
MF	0.567	707	0.629	509
BP	0.520	2026	0.634	2088

Table 16: Comparison of GOstruct and our classifier using the November 2016 GO annotations

between our classifier and GOstruct but our classifier deals with 9 more GO classes. For MF, GOstruct is better by 5% and for BP, by 11%. This tells us that there is more room for improvement for our method when it comes to annotating yeast proteins.

Aside from comparing our method with GOstruct using the annotations from June 2014, we also perform another comparison using more recent human and yeast annotations (November 2016). Table 16 shows the results of this comparison.

We see that some of the same trends from the 2014 annotations still hold true for the 2016 annotations. Specifically, our method performs better on human proteins but not on yeast proteins. However the number of classes for each ontology has increased significantly.

In CC, we have 511 classes (previously, it was 354), that is 166 more classes than GOstruct has. Our F-max for human proteins drops by around 1.6% compared to 2014 but it is still a little bit higher than GOstruct. Meanwhile for yeast proteins, we have the exact same F-max as GOstruct. In MF, we now have 198 more classes than GOstruct but our F-max is still better by around 12% for human proteins. For yeast proteins, GOstruct still has higher F-max. In BP, we have 62 less classes compared to GOstruct and our performance is higher by almost 2% for human proteins while for yeast, GOstruct does better by more than 11%.

5 Summary and Future Work

The aim of this thesis is to explore the use of abstracts of biomedical literature for protein function prediction with the Gene Ontology (GO) by developing a hierarchical classifier that associates features extracted from abstracts with GO terms.

Our hierarchical classifier is composed of a set of independent binary classifiers, each trained to recognise a GO term. We use Naïve Bayes classification method for our base binary classifiers because it is fast to train and has been shown to work well with text classification. To train our binary classifiers, we use the *Less Inclusive* strategy of diving the dataset into positive and negative sets. To extract features from the abstracts, we decompose the abstracts into a bag-of-words representation and compute the TF-IDF vectors for each instance. We use TF-IDF vectors instead of the more common count vectors because the former down-weights common terms and up-weights rarer and more informative terms. This is particularly useful for us since there are many common terms in biomedical literature that are not considered stop words and therefore would not be removed during pre-processing.

We assembled four different datasets for training our classifiers:

1. Curated dataset: composed of abstracts of papers used to support an annotation. These are the papers in the UniProt-GOA annotations.
2. PubMed GO dataset: assembled by searching PubMed using the names of GO terms.
3. PubMed Gene dataset: assembled by searching PubMed using the names of annotated proteins.
4. Curated+PubMed Gene dataset: the union of the Curated and PubMed Gene datasets.

For our experiments, we compare the performance of classifiers trained on data from Curated, PubMed GO, PubMed Gene and Curated+PubMed Gene. All the test instances are drawn from the Curated dataset because these abstracts have been reviewed by human curators to support an annotation and therefore we can be assured of their relevance to that annotation. We use five-fold cross validation to evaluate Curated and Curated+PubMed Gene to avoid overfitting.

We use three different approaches to evaluate our classifier:

1. Paper-centric approach: the goal is for the classifier to predict the GO terms associated with a single abstract.
2. Protein-centric approach: the goal is to predict the GO terms associated with

a protein.

3. Term-centric approach: a complement of the protein-centric approach where the goal is to predict the proteins associated with a GO term.

For our evaluation metrics, we use hierarchical F-max for the paper-centric, protein-centric and term-centric predictions and S-min for the protein-centric predictions. Hierarchical F-max gives the same weight to all GO terms while S-min takes into account the information content (IC) of the terms.

In the paper-centric approach, the best dataset for CC and BP is PubMed Gene. For MF, it is Curated. In this approach, it is better to avoid the largest dataset, Curated+PubMed Gene, because it leads to over-confident predictions (high recall but low precision). CC is the best-performing ontology followed by MF, then BP. We attribute this to CC terms being more unequivocal compared to MF and BP terms and more likely to be found "as-is" in literature.

In the protein-centric approach, we divide proteins into three categories: multi-species proteins, human proteins and yeast proteins. For all three categories and for all three ontologies, Curated+PubMed Gene gives the best F-max, outperforming their component datasets (Curated and PubMed Gene) when they are used individually. We attribute this to the two datasets, assembled using two different methods, complementing each other.

In the term-centric approach, we want to explore the relationship between the F-max of a GO term and characteristics such as IC, closeness centrality and betweenness centrality. We find a negative correlation between F-max and IC where lower IC terms tend to have higher F-max and vice-versa. We find a weak positive correlation between F-max and closeness centrality and a stronger positive correlation between F-max and betweenness centrality.

Lastly, we compare our method with GOstruct, the state-of-the-art literature-based protein function prediction method. Our classifier performs better on human proteins but worse for yeast proteins. The biggest difference we see is in MF for human proteins. This tells us that the binary classifier approach to annotation can perform just as well as the structured output SVM approach used by GOstruct.

Literature-based protein function prediction is not as well-explored as function prediction using other sources such as sequence similarity and interaction networks although biomedical literature data is not meant to replace these methods but to augment them. With that said, there are still many unexplored questions in us-

ing literature data for this task. From the results of this thesis, we suggest some directions for future work in this area:

- Instead bag-of-words features, we can explore the usage of n-grams which can be more suitable in cases where protein names and GO terms are composed of multiple words. These n-grams can also be combined with co-occurence features, similar to GOstruct's co-mentions.

- The threshold where F-measure is maximised is the lowest possible threshold in most of our experiments. This could be because Naïve Bayes probability estimates are not reliable or there is not just enough data for the classifier to make a more confident prediction. Regardless of the reasons, we can mitigate this issue by calibrating our classifiers [ZE01]. Reliable probabilities are important for end-users who would use our classifier as a starting point for their experiments.

- It will be interesting to investigate the feasibility of using other classification algorithms such as support vector machines (SVM) and decision trees as the base binary classifier.

- In our experiments, all abstracts are treated in the same manner. However, it might be useful to investigate the effect of segregating abstracts by publication date or journal. Since protein annotations and GO terms are constantly changing, there might be some terms that have been introduced only recently or terms that used to be popular a few years back but has experienced less usage in recent years. For the former case, a classifier trained only on papers from say, the last five years would be more effective. For the latter, disregarding papers from the last five years would be a more reasonable approach.

- The Human Phenotype Ontology (HPO) is an ontology of human phenotype abnormalities that is structured much like the GO [KDM$^+$13]. Proteins that are associated with a phenotypic abnormality are annotated with the appropriate HPO term. The same concepts and methods that we have used in this thesis can also be applied to predict HPO terms.

6 Acknowledgements

I would like to thank my thesis advisers, Dr. Alan Medlar from the Institute of Biotechnology and Dr. Teemu Roos from the Department of Computer Science for agreeing to supervise this thesis. I would also like to thank the researchers in the Holm Group at the Institute of Biotechnology for giving me the chance to undertake a research project with their group: Dr. Liisa Holm, Dr. Petri Törönen and Dr. Alan Medlar.

References

ABB+00 Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. et al., Gene ontology: tool for the unification of biology. *Nature genetics*, 25,1(2000), pages 25–29.

Bav50 Bavelas, A., Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22,6(1950), pages 725–730.

BDH+09 Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'donovan, C. and Apweiler, R., Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25,22(2009), pages 3045–3046.

Bir06 Bird, S., Nltk: the natural language toolkit. *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pages 69–72.

Bor05 Borgatti, S. P., Centrality and network flow. *Social networks*, 27,1(2005), pages 55–71.

C+15 Consortium, G. O. et al., Gene ontology consortium: going forward. *Nucleic acids research*, 43,D1(2015), pages D1049–D1056.

CAC+09 Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. et al., Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25,11(2009), pages 1422–1423.

CBV10 Cesa-Bianchi, N. and Valentini, G., Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *MLSB*, 2010, pages 14–29.

Con17 Consortium, T. U., Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45,D1(2017), page D158. URL `+http://dx.doi.org/10.1093/nar/gkw1099`.

CR13 Clark, W. T. and Radivojac, P., Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, 29,13(2013), pages i53–i61.

DBF07 DeCoro, C., Barutcuoglu, Z. and Fiebrink, R., Bayesian aggregation for hierarchical genre classification. *ISMIR*. Vienna, 2007, pages 77–80.

DDB11 Dudley, J. T., Deshpande, T. and Butte, A. J., Exploiting drug–disease relationships for computational drug repositioning. *Briefings in bioinformatics*, page bbr013.

DP97 Domingos, P. and Pazzani, M., On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29,2-3(1997), pages 103–130.

dPŠD11 du Plessis, L., Škunca, N. and Dessimoz, C., The what, where, how and why of gene ontology: a primer for bioinformaticians. *Briefings in bioinformatics*, page bbr002.

EPS+05 Eisner, R., Poulin, B., Szafron, D., Lu, P. and Greiner, R., Improving protein function prediction using the hierarchical structure of the gene ontology. *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on*. IEEE, 2005, pages 1–10.

FCV+09 Fontana, P., Cestaro, A., Velasco, R., Formentin, E. and Toppo, S., Rapid annotation of anonymous sequences from genome projects using semantic similarities and a weighting scheme in gene ontology. *PLoS One*, 4,2(2009), page e4619.

FKBHV15 Funk, C. S., Kahanda, I., Ben-Hur, A. and Verspoor, K. M., Evaluating a variety of text-mined features for automatic protein function

prediction with gostruct. *Journal of biomedical semantics*, 6,1(2015), page 9.

Fla12 Flach, P., *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

Fre77 Freeman, L. C., A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

HSMM+15 Huntley, R. P., Sawford, T., Mutowo-Meullenet, P., Shypitsyna, A., Bonilla, C., Martin, M. J. and O'donovan, C., The goa database: gene ontology annotation updates for 2015. *Nucleic acids research*, 43,D1(2015), pages D1057–D1063.

Hun07 Hunter, J. D., Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9,3(2007), pages 90–95.

JOC+16 Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D?Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur, A. et al., An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, 17,1(2016), page 184.

JWHT13 James, G., Witten, D., Hastie, T. and Tibshirani, R., *An introduction to statistical learning*, volume 6. Springer, 2013.

KDM+13 Köhler, S., Doelken, S. C., Mungall, C. J., Bauer, S., Firth, H. V., Bailleul-Forestier, I., Black, G. C., Brown, D. L., Brudno, M., Campbell, J. et al., The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic acids research*, page gkt1026.

KS97 Koller, D. and Sahami, M., Hierarchically classifying documents using very few words. Technical Report, Stanford InfoLab, 1997.

KTNKH15 Koskinen, P., Törönen, P., Nokso-Koivisto, J. and Holm, L., Pannzer: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. *Bioinformatics*, 31,10(2015), pages 1544–1552.

L+98 Lin, D. et al., An information-theoretic definition of similarity. *ICML*, volume 98. Citeseer, 1998, pages 296–304.

MRS+08 Manning, C. D., Raghavan, P., Schütze, H. et al., *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

PVG+11 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pages 2825–2830.

Qui14 Quinlan, J. R., *C4. 5: programs for machine learning*. Elsevier, 2014.

RCO+13 Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A. et al., A large-scale evaluation of computational protein function prediction. *Nature methods*, 10,3(2013), pages 221–227.

Res95 Resnik, P., Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

RSA+07 Ramirez, F., Schlicker, A., Assenov, Y., Lengauer, T. and Albrecht, M., Computational analysis of human protein interaction networks. *Proteomics*, 7,15(2007), pages 2541–2552.

Say09 Sayers, E., The e-utilities in-depth: parameters, syntax and more. *Entrez Programming Utilities Help [Internet]*.

SBH10 Sokolov, A. and Ben-Hur, A., Hierarchical classification of gene ontology terms using the gostruct method. *Journal of bioinformatics and computational biology*, 8,02(2010), pages 357–376.

SBW15 Shatkay, H., Brady, S. and Wong, A., Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics. *Methods*, 74, pages 54–64.

SDRL06 Schlicker, A., Domingues, F. S., Rahnenführer, J. and Lengauer, T., A new measure for functional similarity of gene products based on gene ontology. *BMC bioinformatics*, 7,1(2006), page 302.

SJF11 Silla Jr, C. N. and Freitas, A. A., A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22,1-2(2011), pages 31–72.

SPIBA03 Shah, P. K., Perez-Iratxeta, C., Bork, P. and Andrade, M. A., Information extraction from full text scientific articles: Where are the keywords? *BMC bioinformatics*, 4,1(2003), page 20.

SS08 Schult, D. A. and Swart, P., Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, volume 2008, 2008, pages 11–16.

VRH+87 Vaeck, M., Reynaerts, A., Höfte, H., Jansens, S., De Beuckeleer, M., Dean, C., Zabeau, M., Montagu, M. V. and Leemans, J., Transgenic plants protected from insect attack. *Nature*, 328, pages 33–37.

WS13 Wong, A. and Shatkay, H., Protein function prediction using text-based features extracted from the biomedical literature: the cafa challenge. *BMC bioinformatics*, 14,3(2013), page S14.

WVS+99 Walker, M. G., Volkmuth, W., Sprinzak, E., Hodgson, D. and Klingler, T., Prediction of gene function by genome-scale expression analysis: prostate cancer-associated genes. *Genome research*, 9,12(1999), pages 1198–1203.

WZH05 Wu, F., Zhang, J. and Honavar, V., Learning classifiers using hierarchically structured class taxonomies. *International Symposium on Abstraction, Reformulation, and Approximation*. Springer, 2005, pages 313–320.

ZE01 Zadrozny, B. and Elkan, C., Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *ICML*, volume 1. Citeseer, 2001, pages 609–616.

Zha04 Zhang, H., The optimality of naive bayes. *AA*, 1,2(2004), page 3.