An Empirical Evaluation of the Effectiveness of Different Types of Predictor Attributes in Protein Function Prediction

Fernando Otero[1], Marc Segond[2], Alex A. Freitas[1], Colin G. Johnson[1], Denis Robilliard[2], and Cyril Fonlupt[2]

[1] Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK
{febo2,A.A.Freitas,C.G.Johnson}@kent.ac.uk

[2] Laboratoire d'Informatique du Littoral, ULCO BP719, 62100 Calais, France
{segond,robillia,cyril.fonlupt}@lil.univ-littoral.fr

Summary. Many classification schemes for defining protein functions, such as Gene Ontology (GO), are organised in a hierarchical structure. Nodes near the root of the hierarchy represent general functions while nodes near the leaves of the hierarchy represent more specific functions, giving the flexibility to specify at which level the protein will be annotated. In a data mining perspective, hierarchical structures present a more challenging problem, since the relationship between nodes need to be considered. This chapter presents an empirical evaluation of different protein representations for protein function prediction in terms of maximizing predictive accuracy, investigating which type of representation is more suitable for different levels of the GO hierarchy.

1 Introduction

The recent exponential increase in the number of proteins being identified and sequenced using high throughput experimental approaches has lead to a growth in the number of uncharacterised proteins. Determining protein functions is a central goal of bioinformatics, and it is crucial to improve biological knowledge, diagnosis and treatment of diseases. While biological experiments are the ultimate methods to determine the function of proteins, it is not possible to perform a functional assay for every uncharacterised protein. This is due to time and financial constraints, together with the complex nature of these experiments. Hence, a need for using computational methods to assist the annotation of large amounts of protein data appeared. In particular, this presents a significant opportunity to apply data mining techniques to analyse and extract knowledge from biological databases.

In essence, bioinformatics refers to the research area that combines computational and statistical methods to manage and analyse biological data [11].

It became a very popular research field after the fully sequenced genomes of numerous organisms enabled biologists to map, sequence and analyse individual genes and their protein products. The information acquired by biological experiments has helped to expand understanding about cellular biology, more specifically about biological functions of proteins.

Proteins are large and complex molecules, assembled from amino acids arranged in a linear sequence using information encoded in genes. Proteins perform most of the functions within a cell. For instance, almost all biological processes, including metabolism, need enzymes to catalyse chemical reactions in order to occur; transport proteins are involved in the movement of small molecules through membranes. The amino acid sequence contains all the information necessary to specify the three-dimensional structure of a protein, enabling the protein to perform its function.

Biological databases accumulate vast amounts of protein data, from protein sequences to three-dimensional structures. To facilitate both collaboration and standardization across different sources, biological databases employ controlled vocabularies (ontologies) to annotate protein sequences and features. Ontologies such as the Enzyme Commission (EC) [26], Gene Ontology [2] and SCOP [18] are organised in a hierarchical structure, allowing the annotation of proteins at a different level of detail. In a hierarchical structure, nodes at the top (near the root of the hierarchy) represent general details while nodes at the bottom (near the leaves of the hierarchy) represent more specific details, giving the flexibility to specify at which level the protein will be annotated. The hierarchy defines a parent-child relationship between nodes, where the child is a specialisation of the parent. In a data mining classification task perspective, hierarchical structures present a more challenging problem [8] than flat (single-layer) problems. It is generally more difficult to discriminate between specific classes represented by leaf nodes than more general classes represented by internal nodes, since the number of examples per leaf node tends to be smaller compared to internal nodes.

In this chapter, we apply data mining methods to induce a classification model which can be used to predict the function of uncharacterised proteins using the Gene Ontology functional classification scheme. The Gene Ontology is a complex case of hierarchical organisation, where its terms are arranged in a directed acyclic graph (DAG) structure. We are particularly interested in comparing the effectiveness of different protein representations in terms of maximizing predictive accuracy. Since the problem of discriminating between terms at deeper levels of the hierarchy is different from terms at higher levels, our focus is on investigating which type of representation is more suitable for different GO terms at different levels of the GO hierarchy.

The remainder of this chapter is organised as follows. Section 2 presents a brief introduction of the basic concepts of molecular biology involved in the problem of predicting protein functions. Section 3 describes the methodology for evaluating different protein representations, including data preparation.

Section 4 presents the computational results. Finally, Section 5 presents the conclusion and future research directions.

2 Biological Background

The genetic information of living organisms is stored in DNA (deoxyribonucleic acid) molecules. DNA is a long molecule composed by a sequence of deoxyribonucleic bases, which are linked together by a backbone composed by deoxyribose sugar and phosphate groups. There are four possible nucleotide bases that are found in DNA: adenine (A), guanine (G), cytosine (C) and thymine (T). The DNA molecule structure is a double stranded helix, which is dependent on pairing between the nucleotide bases: adenine is able to pair with thymine, while guanine is able to pair with cytosine. Consequently, the strands in the double helix complement each other in an anti-parallel fashion.

The correlation between genes and proteins is that nucleotide sequences – corresponding to particular genes – in DNA molecules code for amino acid sequences of proteins. This relationship is part of the *central dogma* of molecular biology [1]. The central dogma states that the information flows from DNA to RNA (ribonucleic acid) to protein. In summary, this process works as follows (illustrated in Fig. 1). In the first step (transcription), the genetic information stored in a DNA sequence is used to create a mRNA (*messenger* RNA) molecule. In the second step (translation), this mRNA molecule is used as a template to synthesize proteins. A series of three nucleotides in the mRNA corresponds to a codon, which in turn corresponds to either a specific amino acid or a signal site (start/stop translation). For more details about this process refer to [1].

Proteins are involved in most biological activities and even make up the majority of cellular structures. Since proteins do almost all the work in a cell, understanding the roles of proteins is the key to understanding how the whole cell operates.

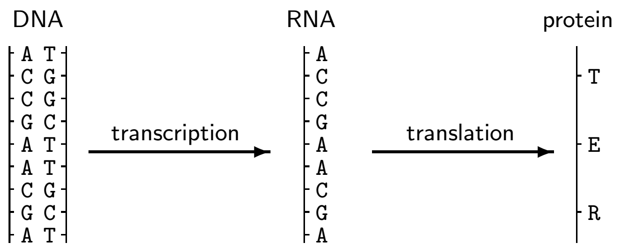

Fig. 1. The central dogma's information flow: from DNA to RNA to protein.

2.1 Proteins

Proteins are the building blocks from which every cell in an organism is built [1]. A protein molecule is assembled from a long sequence of amino acids using information encoded in genes. Each protein has its own unique sequence of amino acids, which is specified by the nucleotide sequence of the gene encoding the protein. There are 20 different types of amino acids, each with different biochemical properties, that can be found in a sequence. An amino acid is composed by a central carbon (C) – α carbon – attached to an hidrogen (H), an amino group (NH_2), a carboxyl group (COOH) and a variable side chain (R). There are 20 distinct side chains, resulting in 20 different types of amino acids. The amino acids are linked together by a peptide bond between their amino and carboxyl groups, constituting the protein's backbone (illustrated in Fig. 2). In general, proteins are 200-400 amino acids long.

The amino acid sequence of a protein is also known as the protein's *primary structure*. It determines the protein's three-dimensional structure and function. Subsequently bondings between the amino and carboxyl groups from different amino acids allow the linear sequence to fold into structures known as alpha helices and beta sheets. Alpha helices (α-helices) are formed when the backbone twists into right-handed helices. Beta sheets (β-sheets) are formed when the backbone folds back on itself in either a parallel or anti-parallel fashion. These structures constitute the protein's *secondary structure*. The three-dimensional shape of the whole protein is known as the protein's *tertiary structure*, which is defined by the spatial relationship between the secondary structures. The three-dimensional shape of a protein is crucial for its function, hence discovering its tertiary structure can provide important information about how the proteins performs its function. Proteins are also capable of assembly into complex structures, known as the protein's *quaternary struc-*

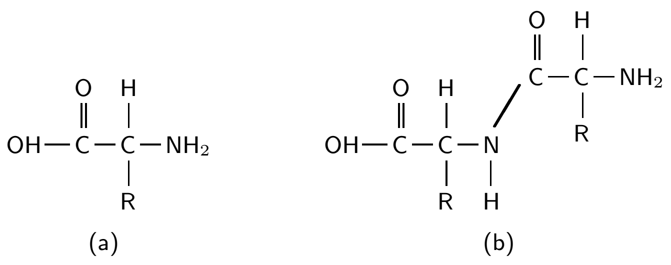

Fig. 2. In (a) basic amino acid structure; (b) The peptide bond between two amino acids (thick line). Amino acids are linked together by a peptide bond between their amino and carboxyl groups, constituting the protein's backbone. This process is repeated many times for polypeptide proteins.

ture, as a result of interaction between them. There are some proteins that can only be functional when associated in protein complexes [15].

The complexity of determining the different levels of protein structures increases from primary towards quaternary structures. For instance, the primary structure can be determined by translating the DNA sequence of the gene that specifies the protein to an amino acid sequence, while the tertiary structure can be determined using complex X-ray crystallography experiments. Consequently, many more proteins sequences (primary structures) are known than proteins three-dimensional structures (tertiary structures). The process from which a protein in one-dimensional state (primary structure) turns to a three-dimensional state (tertiary structure) is called *folding*. While folding occurs spontaneously within a cell, its inherent details are not known.

2.2 Protein Databases

Protein information is widely available in biological databases. Some databases are dedicated to a particular aspect, such as structural information or protein interaction data, while others provide broad information with links to specialised databases. Table 1 presents a summary of biological databases. Most of these databases provide an online search interface.

In general, database entries comprise experimental results combined with annotations. Annotations provide valuable information about a protein, including simple information derived from proteins' primary structures (e.g molecular weight and sequence patterns), nature of the experiment and organism where the protein is found. They can be determined by computational methods or manually, where the latter is more preferable for its reliability. For instance, UniProt (Universal Protein Resource) contains two sections: Swiss-Prot and TrEMBL. Swiss-Prot is the richest annotated protein sequence section, containing manually annotated/curated entries, with extensive database cross-references and literature citations. The TrEMBL section contains computationally analysed records that await full manual annotation.

A commonly used source of protein annotation information are motif databases. Motifs are preserved amino acid sequences, which usually represent a protein family, domain or an activation site. PROSITE [12], PRINTS [3], Pfam [7] and InterPro [17] are examples of databases that contain a collection of protein motifs. Biological literature databases, such as MEDLINE (Medical Analysis and Retrieval System Online), are a valuable resource and textual analysis of these databases is an area of growing interest [22]. More specialised databases contain information about protein interaction data, protein secondary structures, gene expression data, among others.

2.3 Classification Schemes

Several classification schemes for defining protein function annotation exists, such as as the Enzyme Commission (EC) [26] scheme for enzyme classification and FunCat [24] for the functional description of proteins from diverse

Table 1. Summary of biological databases available online.

Name	Description
UniProt http://www.ebi.ac.uk/uniprot/	automatically (UniProtKB/TrEMBL) and manually (UniProtKB/Swiss-Prot) annotated protein sequences
IntAct http://www.ebi.ac.uk/intact/	protein interaction data
CATH http://www.cathdb.info/	hierarchical domain classification of protein structures
PROSITE http://www.expasy.org/prosite/	protein domains, families and functional sites
PubMed http://www.ncbi.nlm.nih.gov/pubmed/	biomedical literature
Pfam http://pfam.sanger.ac.uk/	protein domains and families
InterPro http://www.ebi.ac.uk/interpro/	protein families, domains and sequence patterns
DIP http://dip.doe-mbi.ucla.edu/	protein interaction data
MEDLINE http://medline.cos.com/	biomedical literature
TIGRFAMs http://www.tigr.org/TIGRFAMs/	protein families
PRINTS http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/	protein families
ArrayExpress http://www.ebi.ac.uk/arrayexpress/	gene expression data

organisms. In order to make classification schemes open for computational processing, they usually employ a controlled vocabulary (ontology) to define protein functions. More complex schemes are hierarchically structured, allowing protein annotations at different levels, depending on the depth of knowledge about the protein in question.

The Gene Ontology (GO) Consortium [2] has developed ontologies to classify proteins in terms of three different domains: molecular function, biological process and cellular component. The ontologies are defined by a hierarchy of terms (categories), where each term has a unique numerical identifier and a textual description, arranged in a DAG-like structure. In DAG-based hierarchies, terms can have more than one parent, as opposed to just one parent in tree-based hierarchies. This makes the hierarchical classification problem a particularly challenging one.

Within the GO, the ontology is divided into three different *domains*, each of which consists of a vocabulary for a particular type of biological knowledge. The Molecular Function (MF) domain describes activities performed at the molecular level, generally accomplished by individual proteins. Examples of molecular functions defined are the general concept 'transporter activity' and its specialisation 'ion transporter activity', where the latter is represented as a child of transporter activity. The Biological Process (BP) domain describes activities accomplished by a series of events or molecular functions. Examples of activities defined are high-level processes 'immune response' and 'reproductive process'. Finally, the Cellular Component (CC) domain describes locations, at the levels of subcellular structures and macromolecules complexes. In general, proteins are located in or are a subcomponent of a particular cellular component. Examples of locations defined are 'plasma membrane' and 'golgi transport complex'.

In the GO hierarchy, parent-child relationships are governed by the *true path rule*. The true path rule states that the path from a child term towards top-level terms must always be true. In other words, if a protein is annotated with a term A, it automatically inherits the annotation of all ancestor terms of A. For example, a protein annotated with 'ion transporter activity' will inherit the annotation 'transporter activity', since 'ion transporter activity' is a specialisation of 'transporter activity'. The hierarchical structure allows annotation of proteins at different levels, from general (parent) to more specific (child) terms, depending on the depth of knowledge about the protein in question. The GO classification scheme is currently the preferred approach for computational functional annotation [9].

2.4 Protein Function Prediction

As aforementioned, the exponential increase in the number of proteins being identified and sequenced using high throughput experimental approaches has lead to a growth in the number of uncharacterised proteins (proteins for which the function is unknown). Since the rate at which sequencing methods are producing data is far outperforming the rate at which biological methods can determine protein functions, there is a crescent interest in automated protein function prediction methods.

A commonly used approach is to assign a function by sequence similarity, using BLAST (Basic Local Alignment Search Tool) to perform a similarity search in a protein sequence database. This approach relies on the assumption that proteins with similar sequences perform similar functions. It has been shown that proteins with very different sequences may perform the same function, or proteins with very similar sequences may perform different functions ([10], [27], [25]). Furthermore, this approach is unsuitable if a similar protein with known function cannot be found.

Another approach is to apply data mining techniques to analyse and extract knowledge from biological databases. In this context, an example (record, data instance) represents a protein, the attributes represent different features of a protein (i.e. sequence length, presence/absence of a particular motif) and the classes correspond to the different functions that the protein can perform. For instance, in [13] neural networks were trained using sequence derived features such as amino acid biochemical properties and secondary structure; [15] used protein interaction data to create a probabilistic model based on markov random fields; PROSITE patterns were used in [19] to extract classification rules using C4.5; and [29] followed a clustering approach using protein domains and textual information from MEDLINE. For further examples refer to [23], [5] and [9].

3 Methods

The data mining task addressed in this work is the classification task, where the goal is to predict the class (function) of an example (protein), given the values of a set of attributes for that example. In essence, the classification task consists of inducing a model from the data by observing relationships between predictor attributes and classes, which can be used later to classify new examples.

We have chosen different types of protein representations to be used as predictor attributes (detailed in Subsection 3.2) and a classification algorithm (detailed in Subsection 3.3) to induce a model for protein function classification. We are interested in evaluating those types of representations at different levels of the GO hierarchy, and this evaluation is performed by measuring the predictive accuracy obtained by the same classification algorithm when using each of the different types of protein representation.

3.1 Data Preparation

The selection of the protein examples was divided into three phases. In the first phase we selected a subset of the Gene Ontology hierarchy to represent the classes in our classification problem. As we are interested in ion channel proteins, all the ancestor and descendant terms of the GO:0005216 (*ion channel activity*) node were selected. Note that in the GO hierarchy one term can have more than one parent. For this reason, for every descendant (child) term of the node GO:0005216, we also retrieved its ancestor nodes. The reason for selecting ancestor terms of the GO:0005216 term was to increase the number of negative examples in the data sets. The class hierarchy after this phase was composed by 88 GO terms.

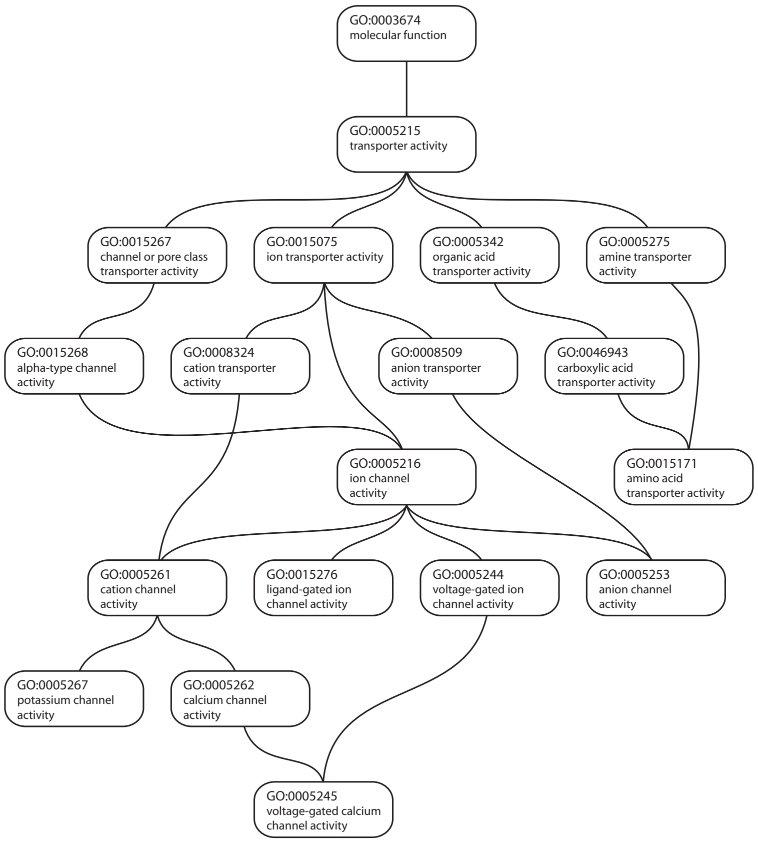
GO:0003674
molecular function
GO:0005215
transporter activity
GO:0015267
channel or pore class transporter activity
GO:0015075
ion transporter activity
GO:0005342
organic acid transporter activity
GO:0005275
amine transporter activity
GO:0015268
alpha-type channel activity
GO:0008324
cation transporter activity
GO:0008509
anion transporter activity
GO:0046943
carboxylic acid transporter activity
GO:0005216
ion channel activity
GO:0015171
amino acid transporter activity
GO:0005261
cation channel activity
GO:0015276
ligand-gated ion channel activity
GO:0005244
voltage-gated ion channel activity
GO:0005253
anion channel activity
GO:0005267
potassium channel activity
GO:0005262
calcium channel activity
GO:0005245
voltage-gated calcium channel activity

Fig. 3. Subset of the Gene Ontology (GO) ion channel hierarchy used in our experiments. GO terms GO:0003674 and G0:0005215 were not used since they represent the root of the hierarchy.

In the second phase, we retrieved protein interaction data from the IntAct database (release 15/12/2007). Records with database cross-references to the GO terms selected in the previous phase were retrieved. In summary, each record of the IntAct database contains the UniProt accession number and a list of interacting proteins which interact with the protein. It should be noted that interacting proteins do not necessarily have to belong to the selected GO hierarchy. It turned out that many GO terms (classes) selected in the previous phase did not have a reasonable number of proteins associated with them. Therefore, we discarded GO terms with less than 10 protein records. At the end of this phase we had selected 147 protein examples and 17 GO terms.

In the third phase, for each protein retrieved in the previous phase we selected the amino acid sequence and MEDLINE document references from the UniProt database (release 12.0). This was accomplished using the database cross-reference to UniProt found in IntAct protein records.

After all three preparation phases, we ended up with a set of 147 protein examples, distributed in 17 GO terms (illustrated in Fig. 3). For each protein, we have retrieved the amino acid sequence, protein interaction data and MEDLINE documents. It should be noted that the number of protein examples selected was constrained by the amount of protein interaction data available. This set of proteins was used to create seven different data sets, as detailed in Subsection 3.2.

3.2 Predictor Attributes

We have selected three different types of protein representations to be used as predictor attributes, namely amino acid composition, protein interaction data and textual information derived from MEDLINE document references.

Amino acid composition attributes were derived directly from proteins' primary sequences by calculating the ratio between the number of amino acid occurrences and the sequence length. For instance, if the amino acid A (Alanine) occurs 14 times in a protein sequence of length 140, the value of attribute A (attribute representing the composition of amino acid A) would be 0.10 (14 / 140). In other words, for each amino acid out of 20 that can be found in a protein sequence, we compute the percentage of the sequence composition relative to the amino acid in question. Using this procedure, 20 numeric attributes were produced for each protein in the data set.

Protein interaction attributes are useful because many proteins interact with one another to perform their function, by assembling multiprotein complexes or metabolic pathways, for example. If one can establish an interaction between a known-function protein and an unknown-function protein, the protein-protein interaction information can be used to predict the function of the unknown-function protein. Protein interaction data was encoded as binary attributes as follows. First, for each protein we retrieved the list of interactor proteins from IntAct database. Then, the complete list of interactors (interactors of all proteins in our data set) was filtered to remove interactors that only interact with one protein in the data set. This restriction was necessary to remove interactors present only in one protein, which do not have any predictive power. Finally, the filtered list of interactors (2095 interactors in total) was encoded as a binary attribute vector. Each position of the vector indicates, with a 'yes' or 'no' value, if the protein interacts with a particular interactor protein or not.

Textual information attributes derived from MEDLINE documents (titles and abstracts) in the form of keywords were encoded as binary attributes, using document references found in proteins' UniProt records. In total, 1010 documents were linked to the 147 proteins of our data set. We applied a Genetic Programming (GP) algorithm, as detailed in Subsection 3.4, to select relevant words (keywords) from linked documents in a preprocessing step. The selected keywords were encoded as a binary attribute vector. Each position of the vector indicates, with a 'yes' or 'no', if the documents linked to the protein contain a particular keyword or not.

Each of the above three types of attributes was used to produce a single data set, namely "AA" (for amino acid composition attributes), "PI" (for protein interaction attributes) and "TX" (for textual information attributes). Furthermore, we explored the combination of predictor attributes in pairs, generating another three data sets ("AA-PI", "AA-TX" and "PI-TX"), and one data set using all types of attributes ("AA-PI-TX"). In total we produced seven data sets, with the same number of examples (as explained in Subsection 3.1) but with a different attribute or combination of attributes in each data set. At the end, two simple predictor attributes derived directly from the proteins' primary sequences, namely **sequence length** and **molecular weight**, were added to each of these seven data sets.

3.3 Classification Algorithm

In order to extract knowledge from the data sets described in the previous Subsection we used the J48 [28] classification algorithm. J48 is a Java implementation of the well known C4.5 decision tree algorithm [21]. We have chosen J48 mainly because C4.5 is a world-class standard induction algorithm and it produces a comprehensible classification model in a decision tree form.

A decision tree consists of internal (decision) nodes, which represent attribute tests, and leaf nodes, which represent classes of the problem in hand. An internal node has outgoing branches, where each branch represents a test outcome value, which in turn connect the nodes of the tree. A leaf node indicates a class to classify an example. Since all nodes in a decision tree have only one parent, with the exception of the root node which has no parent, there is a unique path from a leaf node to the root node. A path can be represented as a conjunction of attribute test outcomes (i.e. all internal nodes and branches followed by the path).

An example is classified by descending the decision tree in a top-down fashion – starting from the root node – following the branches according to the attribute tests' outcomes until a leaf node is reached, where the class associated with the leaf node is assigned to the example. The path followed by the example can be analysed in order to explain the classification of an example into a particular class.

As we are dealing with a class hierarchy represented by a GO subset (illustrated in Fig. 3), and J48 is a flat classifier (i.e., it cannot directly cope with hierarchical classes), we have transformed the hierarchical problem into a set of flat classification problems.[3] The transformation procedure works as follows. For each GO term, we have split the data set into positive (which belong to the GO term) and negative (which do not belong to the GO term) examples. An example (protein) belongs to a specific GO term if it is annotated with that GO term or is annotated with one or more of its child GO terms. For instance, an example annotated with the GO term GO:0015171 will be considered as a positive example for GO terms GO:0015171, GO:0046943, GO:0005275 and GO:0005342 (where the latter three terms are ancestors of the former), according to the class hierarchy. This transformation procedure is required due to the semantics of GO annotation (GO true path rule), where a protein is explicitly annotated only with its most specific GO terms, but it is implicitly considered to have all the ancestral terms of those specific terms [2]. After the transformation step, we trained a classifier for each GO term. That is, each classifier performs a binary classification, predicting whether or not a given example belongs to the classifier's associated GO term.

3.4 Attribute Selection

In order to reduce the number of words to be used as textual information attributes, we performed attribute selection using standard Genetic Programming. Genetic Programming (GP) [14, 4] is an evolutionary technique, based on Darwin's principle of natural selection, which aims at automatically evolving computer programs. In the GP context, a computer program is a solution to the problem at hand, which can be represented as a mathematical equation, a sequence of instructions or an arbitrary combination of input values. GP uses the principle of natural selection to find solutions to complex problems by evolving initially poor solutions into near-optimal ones using a set of genetic operators and a fitness (quality) measure.

Essentially, a GP algorithm consists of a population of candidate solutions to the target problem and an iterative selection process that mimics an evolutionary process. Candidate solutions are selected based on a fitness measure, which measures the quality of candidate solution, to undergo reproduction, recombination and mutation operators in order to form a new population. The new population replaces the old one and a new iteration begins. The fitness-based selection determines that better candidate solutions are more often selected on average, while poor candidate solutions have a smaller change of being selected. New candidate solutions are generated by applying recombination and mutation operators, which are responsible for performing a global search in the solution space. The iterative selection process is carried out until an arbitrary number of iterations (generations of the evolutionary process) is reached or an optimal or satisfactory solution is found.

[3] For a more complete discussion about the differences between flat and hierarchical classification refer to [8].

Table 2. Parameters of the GP algorithm used for attribute selection.

Name	Value
number of generations	50
number of individuals per generation	60000
maximum tree depth	17
mutation rate	5%
crossover rate	90%
reproduction rate (with elitism)	10%

The attribute selection process involves elimination of stop words, word stemming and GP-based selection of predictive words (keywords). It was divided in three steps as follows. In the first step, all MEDLINE documents (titles and abstracts) linked to proteins in our data set were pre-processed by applying the Bow library [16] to carry out stop word removal, followed by stemming using Porter's algorithm [20]. The objective of this step is to remove irrelevant words and to group inflected or derived words to their stem (root form). Examples of stop words are 'a' and 'the', which carry no meaning for text mining purposes. An example of stemming would be to replace the words 'learning' and 'learned' by their stem 'learn'. At the end, a list of words W was generated using all the resultant stems. The second step consists of transforming each document to a vector of word frequencies, using the word list generated in the previous step. For each document d, a vector v_d of length equal to the number of words in W was created where the value of each position i is given by

$$v_d(i) \equiv \frac{\text{number of occurrences of } w_i \text{ in } d}{\text{number of words in } d} , \tag{1}$$

where w_i is the i-th word of W.

In the third step, a GP algorithm was used to identify a list of relevant words for a particular GO term. The selection of words works as follows. Firstly, we transformed the hierarchical classification problem into a set of flat classification problems, using the same procedure described in Subsection 3.3. Then, for each GO term we trained a standard GP to classify the document vectors generated in the previous step. The function set consists of basic arithmetic operators (addition, subtraction and multiplication) and a "max" (maximum value between two numbers) function. The terminal set consists of ephemeral random constants [14] and input nodes representing each dimension of a document vector. The fitness function used is the area under a ROC curve (AUC) [6]. Table 2 presents the parameters of the GP algorithm. At the end, a list of relevant words for each GO term was selected by analysing the five best GP individuals. Words that appear at least in two out of five best individuals were selected as keywords.

4 Computational Results

All experiments were conducted running the well-known 10-fold cross-validation procedure [28]. In essence, a cross-validation procedure consists of splitting the data set into n ($n = 10$, in our case) partitions of approximately same size (number of examples). In an iterative process, each ith ($i = 1, ..., n$) partition was used as the test set and the remaining 9 partitions were temporarily merged and used as a training set. In each iteration, a classification algorithm generates a classification model using the training data. Then, the classification model is used to classify unseen examples from the test set, in order to evaluate the discovered knowledge. The predictive accuracy rate is then computed as the average accuracy rate over the 10 test sets.

The results concerning the predictive accuracy obtained with each type of attribute per GO term are shown in Table 3. An entry in the column "TX" is shown **in bold** if, for the corresponding GO term, the accuracy achieved with textual information attributes is significantly greater than the accuracy achieved with the second best type of attribute (columns "AA" or "PI") for the GO term in question – according to a two-tailed Student's t-test with significance level $\alpha = 1\%$. Overall, the highest predictive accuracy was achieved when using textual information as predictor attributes (data set "TX"). The highest accuracy of "TX" attributes was statistically significant in 7 out of 17 cases. There was no statistically significant difference between the accuracy of "TX" and the accuracy of the other two types of attributes in the remaining 10 cases. These results indicate that textual information attributes are useful for predicting GO terms at any level in the hierarchy used in our experiments. Protein interaction attributes achieved the lowest predictive accuracy in 5 out of 17 cases – in GO terms GO:0015267, GO:0015075, GO:0015268, GO:0008324 and GO:0005216. At the same time, they achieved competitive predictive accuracy (when compared to textual information attributes) in 3 cases – GO terms GO:0008509, GO:0005253 and GO:0005245. These results suggest that protein interaction attributes are useful for predicting GO terms at levels near the leaves of the hierarchy, since at levels near the root of the hierarchy they achieved a significantly lower accuracy. Amino acid composition attributes achieved an average predictive accuracy, overall. In 2 cases – GO terms GO:0008509 and GO:0005253 – the achieved accuracy was competitive with textual information attributes.

The results concerning predictive accuracy per GO term for data sets using a combination of different types of predictor attributes are shown in Table 4. There were no statistically significant differences between the highest predictive accuracy achieved with one combination of attributes and the second best combination of attributes. Also, the combination of all three types of

Table 3. Predictive accuracy (*average* ± *standard deviation*) obtained with each type of attribute per GO term. An entry in the column "TX" is shown **in bold** if, for the corresponding GO term, the accuracy achieved with textual information attributes is significantly greater than the accuracy achieved with the second best type of attribute, columns "AA" (amino acid composition) or "PI" (protein interaction), for the GO term in question – according to a two-tailed Student's t-test with significance level $\alpha = 1\%$.

Term	AA	PI	TX
GO:0015267	69.26 ± 2.01	55.30 ± 3.77	**94.38 ± 1.78**
GO:0015075	71.49 ± 4.37	59.30 ± 1.14	**88.49 ± 2.81**
GO:0005342	84.32 ± 3.01	85.75 ± 2.08	87.18 ± 1.49
GO:0005275	88.57 ± 2.80	87.93 ± 1.53	92.65 ± 1.52
GO:0015268	65.92 ± 3.29	57.02 ± 2.64	**93.24 ± 2.98**
GO:0008324	64.55 ± 3.76	59.92 ± 1.63	**83.77 ± 2.24**
GO:0008509	92.01 ± 1.86	93.21 ± 1.42	93.44 ± 1.87
GO:0046943	86.59 ± 2.38	89.89 ± 1.78	88.55 ± 1.35
GO:0005216	73.32 ± 3.46	57.71 ± 3.60	**93.18 ± 1.77**
GO:0015171	88.59 ± 1.62	89.26 ± 1.74	88.51 ± 2.27
GO:0005261	71.92 ± 2.17	68.68 ± 2.66	**88.93 ± 3.44**
GO:0015276	86.37 ± 2.00	89.18 ± 1.03	85.18 ± 2.34
GO:0005244	69.42 ± 1.95	81.05 ± 3.02	83.58 ± 2.58
GO:0005253	94.03 ± 2.01	94.65 ± 1.85	97.99 ± 1.02
GO:0005267	82.38 ± 1.75	88.52 ± 0.95	**97.29 ± 1.11**
GO:0005262	86.90 ± 2.64	85.64 ± 1.67	87.42 ± 3.15
GO:0005245	89.90 ± 2.84	93.24 ± 2.22	90.57 ± 2.84

predictor attributes did not lead to a significant increase of the predictive accuracy, when compared to the best results using data sets with a single type of attribute. All the combination of attributes that contain textual information attributes achieved competitive predictive accuracy when compared to the predictive accuracy achieved using only textual information attributes ("TX" data set).

Table 5 presents a summary of the results obtained in our experiments. Each cell in that table represent the number of times in which the attribute type in the corresponding row obtained an accuracy significantly greater (positive value) or worse (negative value) than the attribute type in the corresponding column. The value of each cell is in the range $[-17, +17]$, since we are dealing with 17 different GO terms. In 9 out of 17 cases – GO terms GO:0005342, GO:0005275, GO:0008509, GO:0046943, GO:0015171, GO:0015276, GO:0005253, GO:0005262 and GO:0005245 – all single attribute

Table 4. Predictive accuracy (*average* ± *standard deviation*) per GO term for data sets using a combination of different types of predictor attributes. There were no significant differences between the highest predictive accuracy achieved with one combination of attributes and the second best combination of attributes.

Term	AA-PI	AA-TX	PI-TX	AA-PI-TX
GO:0015267	67.38 ± 3.82	94.38 ± 1.78	94.38 ± 1.78	92.95 ± 2.13
GO:0015075	64.17 ± 3.36	87.15 ± 2.30	87.15 ± 2.53	87.87 ± 2.59
GO:0005342	84.32 ± 3.01	87.86 ± 4.23	87.18 ± 1.49	87.86 ± 4.23
GO:0005275	88.57 ± 2.80	93.28 ± 1.99	92.65 ± 1.52	93.28 ± 1.99
GO:0015268	59.20 ± 3.66	88.47 ± 3.19	93.24 ± 2.98	89.90 ± 3.01
GO:0008324	72.54 ± 3.11	85.78 ± 2.08	85.20 ± 2.05	84.49 ± 1.92
GO:0008509	92.01 ± 1.86	94.07 ± 1.48	93.44 ± 1.87	94.07 ± 1.48
GO:0046943	86.59 ± 2.38	86.50 ± 2.16	88.55 ± 1.35	86.50 ± 2.16
GO:0005216	63.12 ± 4.52	91.76 ± 1.73	93.18 ± 1.77	91.76 ± 1.73
GO:0015171	88.59 ± 1.62	89.93 ± 1.76	88.51 ± 2.27	89.93 ± 1.76
GO:0005261	67.71 ± 5.18	86.88 ± 3.56	86.31 ± 2.72	84.88 ± 3.58
GO:0015276	89.18 ± 1.03	86.47 ± 1.68	89.18 ± 1.43	88.52 ± 1.37
GO:0005244	69.42 ± 1.95	84.44 ± 1.33	83.58 ± 2.58	84.44 ± 1.33
GO:0005253	94.03 ± 2.01	95.99 ± 1.79	97.99 ± 1.02	95.99 ± 1.79
GO:0005267	82.38 ± 1.75	96.62 ± 1.13	97.29 ± 1.11	96.62 ± 1.13
GO:0005262	88.10 ± 2.82	86.12 ± 2.86	90.14 ± 3.06	87.55 ± 2.14
GO:0005245	89.90 ± 2.84	89.24 ± 3.00	90.57 ± 2.84	89.24 ± 3.00

Table 5. Summary of the results obtained in our experiments. Each cell represents the number of times in which the attribute type in the corresponding row obtained an accuracy statistically significantly greater (positive value) or worse (negative value) than the attribute type in the corresponding column. The value of a cell is in the range $[-17, +17]$, since we are dealing with 17 different GO terms.

	AA	PI	TX	AA-PI	AA-TX	PI-TX	AA-PI-TX
AA	–	0	-8	0	-8	-8	-8
PI	0	–	-6	0	-6	-6	-6
TX	8	6	–	8	0	0	0
AA-PI	0	0	-8	–	-8	-8	-7
AA-TX	8	6	0	8	–	0	0
PI-TX	8	6	0	8	0	–	0
AA-PI-TX	8	6	0	7	0	0	–

types and combination of types of attributes achieved competitive accuracy, with no significant differences between them. The best results were obtained when using textual information attributes (as a single attribute type or in combination with different types of attributes). Experiments using only protein interaction attributes were slightly better than experiments using only amino acid composition, but both these types of attributes led to poor results, when compared to experiments using textual information attributes.

5 Conclusions and Future Research

This chapter has presented an empirical evaluation comparing the effectiveness of different protein representations in terms of maximizing predictive accuracy using the Gene Ontology (GO) hierarchical functional classification scheme. Hierarchical structures present a challenging problem, since it is generally more difficult to discriminate between specific classes represented by leaf nodes than more general classes represented by internal nodes.

The set of experiments consisted of using different types and combinations of types of predictor attributes for protein function prediction, comparing the predictive accuracy obtained by J48 across a subset of the Gene Ontology ion channel hierarchy. Since J48 is a flat classifier, and we are dealing with a hierarchical classification problem, we have transformed the hierarchical problem into a set of flat classification problems. The results have shown that some types of predictor attributes are more suitable for different levels of the hierarchy. While protein interaction attributes achieved the lowest accuracy at top levels of the hierarchy, they are competitive with the highest accuracy achieved by textual information attributes at lower levels of the hierarchy. Overall, the highest predictive accuracy was achieved when using textual information as predictor attributes, suggesting the importance of using textual information attributes derived from the biological literature for protein function prediction.

As future research, it would be interesting to evaluate different types of predictor attributes, such as gene expression data and post-translational modification data. Also, given the hierarchical nature of predicting function using the GO classification scheme, investigating different measures of predictive accuracy tailored for hierarchical problems may give valuable insights about different protein representations.

Acknowledgements

The authors acknowledge the financial support from an European Union's INTERREG project (Ref. No. 162/025/361). Fernando Otero also acknowledges further financial support from the Computing Laboratory, University of Kent.

References

1. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *The Molecular Biology of the Cell*. Garland Press, 4th edition, 2002.
2. M. Ashburner, C.A. Ball, J.A. Blake, D.Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
3. T.K. Attwood, A. Mitchell, A. Gaulton, G. Moulton, and L. Tabernero. The prints protein fingerprint database: functional and evolutionary applications. In M. Dunn, L. Jorde, P. Little, and A. Subramaniam, editors, *Encyclopaedia of Genetics, Genomics, Proteomics and Bioinformatics*. John Wiley & Sons, 2006.
4. W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone. *Genetic Programming—an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann, 1998.
5. J.R. Bock and D.A. Gough. In Silico Biological Function Attribution: a different perspective. *BioSilico*, 2(1):30–37, 2004.
6. T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2006.
7. R.D. Finn, J. Tate, J. Mistry, P.C. Coggill, S.J. Sammut, H.R. Hotz, G. Ceric, K. Forslund, S.R. Eddy, E.L.L. Sonnhammer, and A.ateman. The pfam protein families database. *Nucleic Acids Research*, 36:D281–D288, 2008.
8. A.A. Freitas and A.C.P.L.F. de Carvalho. A tutorial on hierarchical classification with applications in bioinformatics. In D. Taniar, editor, *Research and Trends in Data Mining Technologies and Applications*. Idea Group, 2007.
9. I. Friedberg. Automated protein function prediction – the genomic challenge. *Briefings in Bioinformatics*, 7(3):225–242, 2006.
10. J.A. Gerlt and P.C. Babbitt. Can sequence determine function? *Genome Biology*, 1(5):1–10, 2000.
11. P.G. Higgs and T. Attwood. *Bioinformatics and Molecular Evolution*. Blackwell Publishing, 2005.
12. N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P. Langendijk-Genevaux, and M. Pagniand C. Sigrist. The prosite database. *Nucleic Acid Research*, 34:D227–D230, 2006.
13. L.J. Jensen, R. Gupta, H.H. Stærfeldt, and S. Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19(5):635–642, 2003.
14. J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
15. S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19(1):i97–i204, 2003.
16. A.K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/∼mccallum/bow, 1996.
17. N.J. Mulder, R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, V. Buillard, L. Cerutti, R. Copley, E. Courcelle, U. Das, L. Daugherty, M. Dibley, R. Finn, W. Fleischmann, J. Gough, D. Haft, N. Hulo, S. Hunter, D. Kahn, A. Kanapin, A. Kejariwal, A. Labarga, P.S. Langendijk-Genevaux, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, C. McAnulla,

J. McDowall, J. Mistry, A. Mitchell, A.N. Nikolskaya, S. Orchard, C. Orengo, R. Petryszak, J.D. Selengut, C.J.A. Sigrist, P.D. Thomas, F.Valentin, D.Wilson, C.H. Wu, and C. Yeats. New developments in the InterPro database. *Nucleic Acid Research*, 35:D224–D228, 2007.

18. A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247:536–540, 1995.
19. G.L. Pappa, A.J. Baines, and A.A. Freitas. Predicting post-synaptic activity in proteins with data mining. *Bioinformatics*, 21(2):ii19–ii25, 2005.
20. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
21. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
22. S. Raychaudhuri. *Computational Text Analysis for Functional Genomics and Bioinformatics*. Oxford University Press, 2006.
23. B. Rost, J. Liu, R. Nair, K.O. Wrzeszczynski, and Y. Ofran. Automatic prediction of protein function. *Cellular and Molecular Life Sciences (CMLS)*, 60(12):2637–2650, 2003.
24. A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and HW. Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acid Research*, 32(18):5539–5545, 2004.
25. W. Tian and J. Skolnick. How well is enzyme function conserved as a function of pairwise sequence identity? *Journal of Molecular Biology*, 333(4):863–882, 2003.
26. E. Webb. *Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology*. Academic Press, 1992.
27. J.C. Whisstock and A.M. Lesk. Prediction of protein function from protein sequence and structure. *Q Rev Biophys.*, 36(3):307–340, 2003.
28. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.
29. H. Xie, A. Wasserman, Z. Levine, A. Novik, V. Grebinskiy, A. Shoshan, and L. Mintz. Large-scale protein annotation through gene ontology. *Genome Research*, 12:785–794, 2002.