Automated Annotation of Keywords for Proteins Related to *Mycoplasmataceae* Using Machine Learning Techniques*

Ana L.C.Bazzan[1†] Paulo M. Engel[1]

Luciana F. Schroeder[1]
[1]Instituto de Informática
Univ. Fed. do Rio Grande do Sul
Caixa Postal 15064
Porto Alegre
91501–970
Brazil
bazzan,engel,luciana@inf.ufrgs.br

Sérgio C. da Silva[2]
[2]Centro de Biotecnologia and Fac. de Veterinária
Univ. Fed. do Rio Grande do Sul
Caixa Postal 15005
Porto Alegre
91501–970
Brazil
ceroni@dna.cbiot.ufrgs.br

January 22, 2003

Abstract

With the increase in submission of sequences to public databases, the curators of these are not able to cope with the amount of information. The motivation of this work is to generate a system for automated annotation of data we are particularly interested in, namely proteins related to the *Mycoplasmataceae* family. Following previous works on automatic annotation using symbolic machine learning techniques, the present work proposes a method of automatic annotation of keywords

*A version of this paper is published in *Bioinformatics 18(S2):S35–S43, October 2002.*
†Corresponding Author

(a part of the SWISS–PROT annotation procedure), and the validation, by an expert, of the annotation rules generated. The aim of this procedure is twofold: to complete the annotation of keywords of those proteins which is far from adequate, and to produce a prototype of the validation environment, which is aimed at an expert who does not have a deep knowledge of the structure of the current databases containing the necessary information s/he needs. As for the first objective, a rate of correct keywords annotation of 60% is reported in the literature. Our preliminary results show that with a slightly different method, applied this method to data related to *Mycoplasmataceae* only, we are able to increase that rate of correct annotation.

1 Introduction

Genome projects have been under a strong pressure to analyze and publish new sequences in the shortest time possible. This way, researchers involved do not have time to complete the annotations; they are providing what seems to be strictly necessary. We see that there is a clear need for automatic tools to generate such an annotation. Following previous works on automated annotation using symbolic machine learning techniques, the present work proposes a method of automatic annotation of keywords (an important field in the SWISS–PROT database). The aim of this procedure is twofold: to complete the annotation regarding keywords (which is far from adequate), and to acquire experience to be able to propose automatic annotation on other (more complex) fields of that database.

There are several networks for genome sequencing and analysis working with different organisms (mostly pathogenic) in Brazil. In [22], the PIGS project was proposed, aiming at providing the infrastructure for the sequencing of the genome of the *Mycoplasma hyopneumoniae*. This bacterium colonizes the respiratory tract of swine and is the primary agent of enzootic pig pneumonia [10, 16], a chronic respiratory disease found at pig farms worldwide that is characterized by high morbidity and low mortality rates. It causes considerable economic losses through retarded growth, poor food conversion, and increased susceptibility of pigs to infection by other organisms [13, 19]. *M. hyopneumoniae* infection is a worldwide problem resulting in economic losses estimated as high as $200 million per year in the US alone. Many studies of *M. hyopneumoniae* have focused on detection of the organism and diagnosis of the disease, because attempts at confining the spread of infection have been unsatisfactory due to the difficulty in detecting the infection early and the lack of effective antimicrobial agents to treat the disease [9, 7]. The diagnosis of *M. hyopneumoniae* is usually done by cultivation of the organism or by immunofluorescence tests performed on frozen thin lung sections with polyclonal antibodies [1, 15]. However, due to the fastidious nature of *M. hyopneumoniae*, its culture and serological identification may take up to one month. Moreover, other mycoplasmas, especially *M. hyorhinis*, easily overgrow and contaminate *M. hyopneumoniae* cultures. Current serological detection methods are further hampered by cross–reactions which have been reported between *M. hyopneumoniae*, *M. hyorhinis*, and *M. flocculare* [2, 8]. Current commercial vaccines are limited in their effectiveness; they do not protect against colonization of the organism,

and outbreaks are occurring with increasing frequency. The disease is one of the most relevant occuring in swine herds in southern Brazil.

Mycoplasmas belong to the *Mycoplasmataceae* family, of which 102 species affect humans and 6 affect animals. Genomes of these organisms vary between 580 and 1.350 Kb [18].

One of the expected results coming from the PIGS project is to fully sequence and annotate the genome of that microorganism. Then, in a later phase, important proteins will be expressed aiming at developing diagnostic tests and vaccine production.

This paper is organized as follows. The next section discusses background ideas concerning automated annotation using machine learning and agent–based techniques. Then our approach is fully detailed. After, we discuss the results achieved so far, and make a comparison of our environment to others, focussing on the indication of further directions regarding our work.

2 Background

There has been an explosion of data, information and computational tools coming out of the diverse genome projects. NCBI and EBI alone report huge databases, mostly not clearly structured. We discuss here the use of technologies of agents (originally developed with other goals in mind) and machine learning in bioinformatics. These are useful because the motivation for their usage in other fields is the same as in bioinformatics: the necessary data is distributed among several sources, it is dynamic, its content is heterogeneous, and most of the work can be done in a parallel way. Hence, *information agents* can integrate multiple distributed heterogeneous information sources. Moreover, with minor changes, agents can handle machine learning techniques automatically.

It is important to notice that there are only a few truly multi–agent projects in the domain of bioinformatics. For instance, InfoSleuth [3] has been used to annotate livestock genetic samples. The scientific community on agents and multi–agent systems is now turning its attention to a key issue to bioinformatics as well: how to ensure the semantic consistency of the integrated data. This is one of the concerns of the Geneweaver project described below, together with K. Decker's framework, and with the MASKS environment.

In [5], a prototype is described whose aim is to automate the annotation of a sequence of a virus. Their work is based on information gathering: search, filtering, integration, analysis, and presentation of the data to the user. It uses the author framework DECAF, a multi-agent system toolkit based on RETSINA [6]. The system has four overlapping multi–agent organizations. The first, Basic Sequence Annotation, is charged with integrating remote gene sequence annotations from various sources with the gene sequences at the Local Knowledge Base Management Agent (LKBMA). The second, Query, allows complex queries on the LKBMAs via a web interface. The third, Functional Annotation, is responsible for collecting information needed to make an informed guess as to the function of a gene, specifically using the three–part Gene Ontology[1].

[1] http://www.geneontology.org/

GeneWeaver [4] is a multi–agent system for managing the task of genome analysis. Since all processes of identifying genes and predicting function of proteins (despite being labour–intensive and requiring expert knowledge) are computer–based tasks, it is possible to automate them. In case of Geneweaver this has been done using a multi–agent approach.

GeneWeaver comprises a community of agents that interact with each other, each performing some distinct task, in an effort to automate the processes involved in, for example, determining protein function. Agents in the system can be concerned with the management of the primary databases, performing sequence analyses with existing tools, or storing and presenting resulting information.

The third agent–based tool we want to tackle here is the MASKS environment [20], whose aim is to improve symbolic learning through knowledge exchange. The motivation of this work is to mimic human interaction in order to reach better solutions. This aims at supporting a recent practice in data mining which is the use of collaborative systems. These are necessary because even if data mining is a powerful technique for knowledge extraction, none of the embbeded algorithm is good in all possible domains. Each algorithm contains an explicit or implicit bias that leads it to prefer certain generalizations over others. Therefore different data mining techniques applied to the same dataset hardly generate the same result. In general, combining inductors increases the accuracy by reducing the bias. This integration aims at overcoming limitations of individual techniques through hybridization or fusion of various techniques. These ideas have led to the emergence of many different kinds of system architecture.

As to what regards machine learning techniques, these have been widely used in bioinformatics as for instance in [12, 11]. Automatic annotation and machine learning are combined in [14]. This work describes a machine learning approach to generate rules based on already annotated keywords of the SWISS–PROT database. Such rules can then be applied to yet unannotated protein sequences. Since this work has actually motivated ours, we provide here just a brief introduction. Details can be found in [14].

In short, the authors have developed a method to automate the process of annotation regarding keywords in SWISS–PROT, based on an algorithm called C4.5 [17]. This algorithm works on training data (in this case, previously annotated keywords regarding proteins). Such data comprises mainly taxonomy entries, INTERPRO classification, and PFAM and PROSITE patterns. Given these data (called attributes), C4.5 derives a classification for a target class (in this case, the keyword). Since dealing with the whole data in SWISS–PROT at once would be prohibitive, the authors divided it in protein groups according to the INTERPRO classification. Then each group was submitted to an implementation of C4.5 contained in the software package Weka [2]. Rules were generated and a confidence factor for each rule was calculated, as it can be seen in Figure 1 (specific case of keyword “Cell division”. Confidence factors were calculated based on the number of false and true positives, by performing a cross–validation, and by testing the rate of error in predicting keyword annotation over the TrEMBL database. The resulting framework (called Spearmint) can be accessed at http://www.ebi.ac.uk/spearmint. If the user clicks on a given rule, a new window appears (Figure 2)

[2] http://www.cs.waikato.ac.nz/~ml/weka

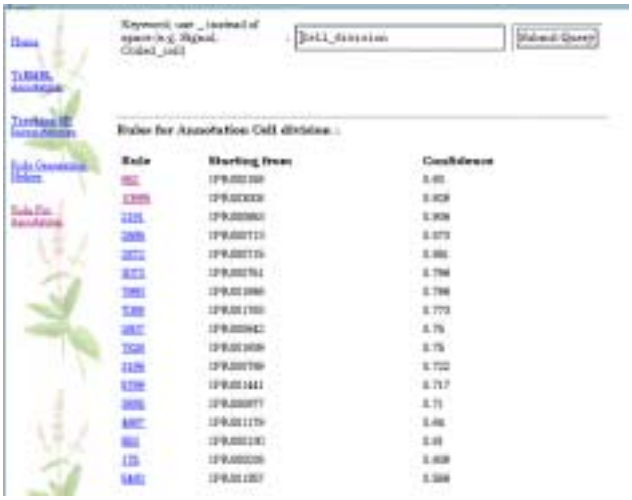

Figure 1: Spearmint web interface: all rules generated for the keyword “Cell division”

```
J48 pruned tree
------------------

IPR000150 = yes: add Keyword 'Cell division' (51.0)

Number of Leaves  :      1

Size of the tree :      1

------------------------------------------------------------

True  Positives          : 51
True  Negatives          : 0
False Positives          : 0
False Negatives          : 0
```

Figure 2: Spearmint web interface: rule number 652 for the keyword “Cell division”

where the details of the rule can be seen, i.e. the data computed by the C4.5 algorithm implemented in Weka.

3 Approach and Results

In this section we describe our approach to tackle the field “Keywords” in the SWISS–PROT database. This is a very important one, used mainly when a researcher wants to compare an unknown sequence s/he is working with, to the sequences already deposited in that database. Unfortunately, regarding the family of *Mycoplasmataceae*, a high number of proteins in SWISS–PROT is classified as “hypothetical protein” (around 50% of them according to data obtained in February 2002). Besides, the proteins in TrEMBL, which are also potential targets for comparison, are poorly annotated regard-

```
SWISSPROT:FTSH_MYCGE;P47695;Cell divisionATP−binding
TransmembraneHydrolaseMetalloproteaseZincComplete proteome;
IPRMATCHES:P47695;IPR000642IPR003593IPR003959IPR003960;
PROSITE:AAA;PS00674;
SWISSPROT:FTSH_MYCPN;P75120;Cell divisionATP−binding
TransmembraneHydrolaseMetalloproteaseZincComplete proteome;
IPRMATCHES:P75120;IPR000642IPR003593IPR003959IPR003960;
PROSITE:AAA;PS00674;
SWISSPROT:AMPA_MYCGE;P47631Q49371;HydrolaseAminopeptidase
ManganeseComplete proteome;IPRMATCHES:P47631;IPR000819;
PROSITE:CYTOSOL_AP;PS00631;
SWISSPROT:AMPA_MYCPN;P75206;HydrolaseAminopeptidaseManganese
Complete proteome;IPRMATCHES:P75206;IPR000819;
PROSITE:CYTOSOL_AP;PS00631;
SWISSPROT:AMPA_MYCSA;P47707;HydrolaseAminopeptidaseManganese;
IPRMATCHES:P47707;IPR000819;PROSITE:CYTOSOL_AP;PS00631;

   [...]
```

Figure 3: Data extracted from SRS (view associating SWISS–PROT, IPRMATCHES, and PROSITE); columns (semi-colon delimited) show: SWISS–PROT code, accession number, keywords list, IPRMATCHES code, INTERPRO accession number, PROSITE code, and PROSITE accession number.

ing the Keywords field[3].

Therefore, the good results achieved in [14] (reported in the previous section) have motivated us to conduct a similar study aiming at automated rule annotation of keywords for the *Mycoplasmataceae* universe of proteins. This way, we can extend the annotation in both TrEMBL and SWISS–PROT for internal use in the PIGS project.

We have initiated by reproducing the approach described in [14]. Soon we realized that since we reduced our universe to the proteins related to *Mycoplasmataceae*, we could do better by modifying their method. Indeed, their method is based on a partition of the SWISS–PROT proteins by INTERPRO Accession Number (henceforth IPR Acc). Thus, rules (to recommend or not the annotation of a keyword) are generated for each IPR Acc (when applicable) and, after that, they are ranked by confidence factor (CF). This may be confusing for the user. For instance, when two or more rules have close CFs, but nonetheless recommend inconsistent annotation (i.e. one does recommend the annotation while another does not), how should the user decide?

The approach we propose here is similar but consider all applicable IPR Acc's as attributes at once, together with PROSITE Accession Number (henceforth PS Acc's). Of course, taxonomy is no attribute in our case since we are dealing with a single family, namely the *Mycoplasmataceae*.

The first data collection was done in February 2002 by means of the SRS web site [4]. Basically, we have performed a query on the SWISS–PROT database in which the field "Organism" included *M. pneumoniae*, an organism close to *M. hyopneumoniae*. After, we have created a view for the SWISS–PROT database which associates this database with the PROSITE and with the IPRMATCHES (through a personal communication

[3] As it can be seen in Table 4, in the data we collected, 378 out of 1894 had no keyword at all, while 896 had no attributes such as INTERPRO classification.

[4] version 6 at www.srs.ebi.ac.uk

with SRS maintainers, we found out that the association with the INTERPRO was not working properly so we used IPRMATCHES instead, which provides the required data as well). This view included:

1. for SWISS–PROT: AccNumber, keywords ;
2. for IPRMATCHES: IPR AccNumber ;
3. for PROSITE: PS AccNumber .

We have decided not to work with PFAM classification since in the TrEMBL database virtually no such classification is annotated for the family we are interested in.

Such procedure has retrieved 1539 proteins (February 2002), as it can be seen in Table 1. There were around 500 IPR Acc's and around 200 PS Acc's related to that organism (actually 714 in total). Around 130 different keywords appeared in the data. The next step was the retrieving process, which is not difficult using SRS. This has generated a table (fields are delimited by ";") partially depicted in Figure 3.

Table 1: First training dataset extracted from SWISS–PROT

Total proteins SWISS–PROT	1539
Total attributes	714

However, this dataset included data which was later considered as noise. For instance, the keyword "Complete proteome" is obviously of no interest in an automatic annotation. Also, it is desirable to exclude hypothetical proteins.

Therefore, a second data collection was done aiming at filtering out proteins which had "Hypothetical protein" or "Complete proteome" as keyword. In order to increase the training set, we decided to retrieve data related to all *Mycoplasmataceae* family instead of data related only to the *M. pneumoniae*. This has yielded the training set shown in Table 2. This way, the number of proteins related to the *Mycoplasmataceae* family decreased to 786 (Feb. 2002) while there were 807 IPR Acc's. This number has increased due to the increasing in the quantity of organisms. The number of different keywords has remained almost the same.

Table 2: Second training dataset extracted from SWISS–PROT

Total proteins SWISS–PROT	786
Total attributes	807

Since we are using the Weka machine learning package to generate the rules, in both cases discussed above we had to transform the datasets provided by SRS into the Weka format, a task which is time–consuming: for instance, for the 1539 lines or proteins and the 714 attributes present in the first dataset, the generation of a single file regarding a keyword took around 50 minutes in a Celeron processor (640 Mhz) using the interpreted programming language of the MATLAB toolbox. A better processor and a compiled language would probably reduce this, although the main problem seems

```
@relation Cell_division

@attribute IPR000642 {TRUE, FALSE}
@attribute IPR003593 {TRUE, FALSE}
@attribute IPR003959 {TRUE, FALSE}
@attribute IPR003960 {TRUE, FALSE}
@attribute IPR000819 {TRUE, FALSE}
@attribute IPR001687 {TRUE, FALSE}
...
@attribute IPR001162 {TRUE, FALSE}
@attribute IPR004791 {TRUE, FALSE}
@attribute Cell_division {yes, no}

@data

TRUE,TRUE,TRUE,TRUE, ... ,FALSE,FALSE, yes
...
```

Figure 4: Input file to Weka (in this case, the file refers to keyword "Cell division")

to be the fact that there are many operations on string manipulation and I/O to files. Therefore, it remains to be tested whether more efficient languages such as C^{++} would do better. Probably, script–languages such as Perl or Python would not provide any help in this case since they are interpreted languages as well.

A typical input file (from the first dataset) is partially depicted in Figure 4. The first line indicates the relation (keyword). Then, 714 lines follow indicating how the attributes are mapped for the 1539 proteins. In our cases, all attributes can only have the values "true" or "false" meaning that either it appears in the classification of a given protein or not. The last of those 714 lines is the target attribute or class ("Cell division" in Figure 4). We opted to map the keyword to "yes" or "no", meaning that either the keyword shall be annotated or not. Finally, there come 1539 lines. Each of these is formed by 714 entries separated by comma.

For instance, in Figure 4 we see that the attribute IPR000642 can be either "TRUE" or "FALSE" (no matter which keyword). In particular, for the keyword "Cell division", that attribute is "TRUE", as one can observe in the first line after the "@data" line in that figure.

To save time, we have generated the Weka rules only for the keywords which appear in valid lines of the test data set (i.e. those from TrEMBL). We have used the C4.5 tool of Weka, but preliminary tests indicate that the algorithm ID3 would do as good. A valid line has to have at least a keyword and at least an attribute IPR Acc or PS Acc. A rule is depicted in Figure 5, where "|" means a conjunction. This rule is to be read as follows. If the protein has the INTERPRO classification IPR000158, then it shall have the keyword "Cell division". But, if the protein does not have IPR000158, then, if it has the IPR001179, then it shall have the keyword "Cell division" as well. Similar boolean tests go on throughout the rule. The last test finally prescribes whether or not a given protein should have the keyword or not: if all tests fail, then the protein shall

```
Relation:     Cell_division
Instances:    1539
Attributes:   714
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

IPR000158 = TRUE: yes (3.0)
IPR000158 = FALSE
|   IPR001179 = TRUE: yes (3.0)
|   IPR001179 = FALSE
|   |   IPR000642 = TRUE: yes (2.0)
|   |   IPR000642 = FALSE
|   |   |   IPR000897 = TRUE
|   |   |   |   IPR004125 = TRUE: no (3.0)
|   |   |   |   IPR004125 = FALSE: yes (3.0)
|   |   |   IPR000897 = FALSE: no (1525.0/3.0)

Number of Leaves  : 6

Size of the tree : 11
```

Figure 5: Rule generated by Weka (keyword “Cell division”)

```
SPTREMBL:O69459;O69459;Signal;IPRMATCHES:O69459;IPR002520
SPTREMBL:O30704;O30704;LipoproteinSignal;IPRMATCHES:O30704;IPR001087
SPTREMBL:O30922;O30922;Lipoprotein;;
SPTREMBL:O52311;O52311;Lipoprotein;IPRMATCHES:O52311;IPR003760
SPTREMBL:Q59527;Q59527;Isomerase;IPRMATCHES:Q59527;IPR002205
SPTREMBL:Q50367;Q50367;Lipoprotein;;
SPTREMBL:Q50368;Q50368;Lipoprotein;;
SPTREMBL:Q9RGX3;Q9RGX3;LipoproteinSignal;IPRMATCHES:Q9RGX3;IPR003760
SPTREMBL:O06764;O06764;LipoproteinSignal;;
SPTREMBL:Q9L8W1;Q9L8W1;LipoproteinSignal;;
SPTREMBL:Q9L8W0;Q9L8W0;LipoproteinSignal;;
SPTREMBL:Q9L8V9;Q9L8V9;LipoproteinSignal;;
SPTREMBL:O53100;O53100;LipoproteinSignal;IPRMATCHES:O53100;IPR002520
SPTREMBL:O53101;O53101;LipoproteinSignal;IPRMATCHES:O53101;IPR002520
SPTREMBL:Q9Z3D4;Q9Z3D4;;IPRMATCHES:Q9Z3D4;IPR001419;
SPTREMBL:Q9Z3D1;Q9Z3D1;;;
SPTREMBL:Q9Z356;Q9Z356;;;
SPTREMBL:Q49468;Q49468Q53303;Signal;;
SPTREMBL:Q9R413;Q9R413;;IPRMATCHES:Q9R413;IPR002019;
SPTREMBL:Q9R3Z9;Q9R3Z9;;;
SPTREMBL:Q9R3X1;Q9R3X1;;;
SPTREMBL:Q9R3N6;Q9R3N6;LipoproteinSignal;IPRMATCHES:Q9R3N6;IPR003760;
SPTREMBL:Q9R3J2;Q9R3J2;;IPRMATCHES:Q9R3J2;IPR001924;
SPTREMBL:Q9R3F3;Q9R3F3;;;
SPTREMBL:Q9R4I6;Q9R4I6;;;
SPTREMBL:Q9R4I5;Q9R4I5;;;
SPTREMBL:Q9R4C8;Q9R4C8;;;
SPTREMBL:O86305;O86305;Cell divisionGTP−binding;IPRMATCHES:O86305;IPR000158IPR003008
...
```

Figure 6: Data extracted from SRS (view associating TrEMBL and IPRMATCHES); columns (semi-colon delimited) show: TrEMBL code, accession number, keywords list, IPRMATCHES code, and INTERPRO accession number

not be annotated with the keyword.

Once the rules were generated, we have proceeded to the evaluation of the quality of these rules. We have performed two kinds of validation: the standard ten–fold cross–validation on the training data (see results on Table 3 and the discussion in next section), and a prediction test on data from another source. Here, the obvious candidate is a dataset extracted from the database TrEMBL, which has a structure similar to SWISS–PROT. The main difference is that TrEMBL has a poorer annotation. However, the existing annotation of keywords is enough for evaluation purposes. The data was extracted from TrEMBL in the way already explained regarding the extraction of data from the SWISS–PROT database (i.e. query, view and save procedures as explained). As it can be seen in Figure 6, many proteins have neither a keyword nor an attribute. Therefore, these lines were deleted.

For those remaining, the following procedure of evaluation was performed: if the protein is annotated with keyword K, then the rule for K (generated by Weka) is checked. For instance, take again the rule in Figure 5. It says (first boolean test) that “if the protein has the INTERPRO classification IPR000158, then it shall have the keyword “Cell division”. As we see in the last line shown in Figure 6, this is in fact the case for protein “O86305” and keyword “Cell division”.

This procedure was repeated for each protein and for each of its keywords (total of around 246 checks in the first dataset and 456 in the second one).

Table 3: Accuracy and Confidence for All Classes after Cross Validation

Class	Accuracy	Confidence	Class	Accuracy	Confidenc
Acyltransferase	99.24	98.34	Membrane	96.56	95.0
Amino–acid_biosynthesis	99.75	99.08	Metal–binding	99.24	98.3
Aminoacyl–tRNA_synthetase	99.87	99.28	Metalloprotease	99.49	98.7
Aminopeptidase	99.75	99.08	Methionine_biosynthesis	99.75	99.0
Antibiotic	99.62	98.88	Methyltransferase	98.86	97.8
Antibiotic_resistance	99.62	98.88	mRNA_processing	99.75	99.0
Antigen	99.11	98.17	Multifunctional_enzyme	99.49	98.7
Arginine_metabolism	99.62	98.88	NAD	98.22	97.0
Aspartyl_protease	99.75	99.08	NADP	99.62	98.8
ATP_synthesis	100.00	99.51	Nickel	99.11	98.1
ATP–binding	97.58	96.26	Nuclease	99.24	98.3
Carbohydrate_metabolism	99.75	99.08	Nucleotide_biosynthesis	99.49	98.7
Cell_adhesion	99.75	99.08	Nucleotidyltransferase	99.36	98.5
Cell_division	99.36	98.52	One–carbon_metabolism	99.24	98.3
CF(0)	99.87	99.28	Oxidoreductase	97.84	96.5
CF(1)	100.00	99.51	Pentose_shunt	100.00	99.5
Chaperone	98.47	97.35	Peptide_transport	98.98	98.0
Cobalt	100.00	99.51	Phosphate_transport	99.49	98.7
Coenzyme_A_biosynthesis	99.75	99.08	Phospholipid_biosynthesis	99.24	98.3
Coiled_coil	99.49	98.70	Phosphorylation	98.47	97.3
Complete_proteome	85.37	82.73	Phosphotransferase_system	98.73	97.6
Cytadherence	98.22	97.03	Primosome	99.75	99.0
DNA_condensation	99.87	99.28	Protein_biosynthesis	97.33	95.9
DNA_damage	99.75	99.08	Protein_transport	99.36	98.5
DNA_recombination	98.98	98.00	Purine_biosynthesis	99.75	99.0
DNA_repair	97.84	96.56	Purine_salvage	99.75	99.0
DNA_replication	98.22	97.03	Pyridoxal_phosphate	99.75	99.0
DNA_synthesis	99.75	99.08	Pyrimidine_biosynthesis	99.75	99.0
DNA–binding	98.73	97.67	Redox–active_center	99.87	99.2
DNA–directed_DNA_polymerase	99.62	98.88	Repeat	97.07	95.6
DNA–directed_RNA_polymerase	98.60	97.51	Repressor	100.00	99.5
Electron_transport	100.00	99.51	Restriction_system	99.62	98.8
Elongation_factor	99.75	99.08	Ribosomal_protein	95.80	94.1
Endonuclease	99.49	98.70	RNA–binding	97.33	95.9
Excision_nuclease	99.49	98.70	Rotamase	100.00	99.5
Exonuclease	100.00	99.51	rRNA_processing	99.75	99.0
FAD	99.36	98.52	rRNA–binding	98.35	97.1
Fatty_acid_biosynthesis	99.75	99.08	Schiff_base	100.00	99.5
Flavoprotein	98.86	97.84	Septation	99.62	98.8
FMN	99.75	99.08	Serine_protease	99.75	99.0
Galactose_metabolism	99.75	99.08	Sigma_factor	99.75	99.0
Gluconeogenesis	100.00	99.51	Signal	97.58	96.2
Glycerol_metabolism	99.75	99.08	Signal_recognition_particle	100.00	99.5
Glycolysis	97.07	95.65	SOS_response	98.73	97.6
Glycosidase	99.75	99.08	Structural_protein	98.98	98.0
Glycosyltransferase	99.49	98.70	Sugar_transport	98.47	97.3
GTP–binding	99.24	98.34	Thiamine_biosynthesis	99.75	99.0
Heat_shock	99.49	98.70	Thiamine_pyrophosphate	99.75	99.0
Helicase	99.24	98.34	Topoisomerase	100.00	99.5
Histidine_biosynthesis	99.75	99.08	Transcription	98.60	97.5
Hydrogen_ion_transport	99.87	99.28	Transcription_regulation	99.62	98.8
Hydrolase	94.40	92.57	Transcription_termination	99.75	99.0
Initiation_factor	100.00	99.51	Transferase	91.35	89.1
Iron	99.49	98.70	Translocation	99.36	98.5
Isomerase	98.73	97.67	Transmembrane	97.20	95.8
Kinase	96.31	94.75	Transport	98.47	97.3
Ligase	98.60	97.51	tRNA_processing	99.36	98.5
Lipid_synthesis	99.75	99.08	tRNA–binding	99.87	99.2
Lipid–binding	100.00	99.51	Virulence	99.62	98.8
Lipoprotein	98.35	97.19	Zinc	97.58	96.2
Lipoyl	99.75	99.08	Zinc–finger	99.62	98.8
Lyase	99.49	98.70	**Average all classes**	**99.02**	**98.1**
Magnesium	98.35	97.19	**Maximum**	**100.00**	**99.5**
Manganese	99.24	98.34	**Minimum**	**85.37**	**82.7**

4 Results and Discussion

We are now in position to discuss the results achieved with the procedure explained in the last section, and to present a comparison to results achieved by similar approaches.

We start with the standard ten–fold cross–validation (Table 3). This method yields accuracy for each class (keyword), computed by the number of true positives (TP) divided by the total of TP plus false positives (FP), i.e. $TP/TP+FP$. We see that the computed accuracy for most of the classes is above 80%.

As for confidence (third and sixth columns in Table 3), we have employed the same formula used in [14]:

$$c = \frac{p + \frac{z^2}{2n} - z\sqrt{\frac{p}{n} - \frac{p^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \tag{1}$$

Due to the nature of the training set, the support for each rule (number of instances on which the algorithm has based the generation of the rule) is very low. Of course, this is a shortcoming but it seems no to influence the validity of the rules generated. The Spearmint tool discussed before also suffers from the same problem (see Figure 2 which shows one rule regarding the keyword "Cell division"). In spite of this shortcoming, as we will see next, the rule generated successfully predicted more than 60% of the keywords in TrEMBL.

Table 4: First dataset used to perform the prediction test; extracted from TrEMBL

Total proteins TrEMBL	1139
Total pairs protein–keyword	1874
Pairs correctly predicted	182 (74.0%)
Pairs incorrectly predicted	64 (26.0%)
Proteins without keyword	378
Proteins without attribute	896
Proteins without keyword or attribute	1035
Proteins without both keyword and attribute	239

Table 5: Second dataset used to perform the prediction test; extracted from TrEMBL

Total proteins TrEMBL	672
Total pairs protein–keyword	456
Pairs correctly predicted	274 (60.1%)
Pairs incorrectly predicted	181 (39.9%)
Proteins without keyword	109
Proteins without attribute	92
Proteins without both keyword and attribute	196

The results regarding the prediction using the TrEMBL data are shown in Table 4 and in Table 5, for the complete and the filtered dataset respectively. In the first case, we observe that around 75% of the keywords were correctly predicted by the method, while there was 25% of false negatives (i.e. the keyword was annotated but could not be predicted by the rule). This rate of prediction is inferior when we tackle the second dataset because this includes proteins related to more organisms of the family

Figure 7: NCBI entry NP_326457.1

Figure 8: Description of motif

Mycoplasmataceae, but less instances due to the filtering applied over the dataset to eliminate the keywords discussed in the last section.

Moreover, for new protein sequences (just released in the PIGS project) the rules predicted keywords which were recognized as correct by an expert. For instance, one sequence has high similarity to that (partially) depicted in Figure 7. The next step was to search the motifs for the sequence in the INTERPRO database, which has indicated that it belongs to the family IPR001687. Now, if we apply the rules generated for each keyword, we conclude that the keyword "ATP binding" should be annotated. An expert has concluded that this annotation was correct. Also the data depicted in Figure 8 demonstrates that.

It is also possible to compare the structure of these rules to the similar ones produced by the Spearmint tool at the web site given in the background section.

5 Conclusion and Future Work

This paper discusses an alternative usage of the method proposed in [14]. Their method is based on the partition of SWISS–PROT by IPR Acc's with the side effect that too many rules are generated. Since we are interested in the annotation of keywords for proteins related to the *Mycoplasmataceae* family, we propose the generation of rules based on a reduced set of proteins extracted from SWISS–PROT. Given this reduced set, it is possible to consider all attributes at once, in a different way of that proposed by [14].

Our method generates a single rule for each keyword, thus avoiding inconsistencies in the proposed annotation. Of course, a similar method can be used for other families of organisms or other partitions of SWISS–PROT. The rules were evaluated using a set of proteins from the TrEMBL database. Results show that the quality of annotation is satisfactory: between 60% and 75% of the given keywords were correctly predicted.

From now on, it is possible to apply the rules for the unknown proteins which are being sequenced in the PIGS project. One has first to annotate the attributes (e.g. classify each protein in IPR and/or PS families) and then apply the rules.

In the future, we want to act in three different directions. First, obtaining better quality regarding the training data. This can be achieved by not considering proteins with undesirable characteristics. Second, we want to extend this method to include other machine learning techniques which can be handled by agents as discussed in the background section. In [21] we compare symbolic methods as the one tackled in the present paper to a neural network model. For the same datasets used here, we have concluded that the neural network output is more compact. However, as one of our goals is to integrate this methods in an environment for automatic annotation of ORFs and proteins, it is important that the human expert be able to "see" the rules generated. This is the biggest disadvantage of the neural network since the rules are not straightforward to the expert. Finally, a third direction is the extension to automated annotation of other fields of the SWISS–PROT database.

6 Acknowledgements

We are greatful for the support of CNPq to the authors A.L.C. Bazzan and L.F. Schroeder, as well as to the PIGS project. We also want to thank the anonymous reviewers of this paper for their suggestions.

References

[1] W. Amanfu, C. N. Weng, R. F. Ross, and H. J. Barnes. Diagnosis of mycoplasmal pneumonia of swine: sequential study by direct immunofluorescence. *Am. J. Vet. Res.*, 45:1349–1352, 1984.

[2] C. H. Armstrong, M. J. Freeman, and L. Sands-Freeman. Crossreactions between *Mycoplasma hyopneumoniae* and *Mycoplasma flocculare*: practical implications for the serodiagnosis of mycoplasmal pneumonia of swine. *Isr. J. Med. Sci.*, 23:654–656, 1987.

[3] R. J. Bayardo, Jr., W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk. InfoSleuth: Agent-based semantic integration of information in open and dynamic environments. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 195–206, New York, 1997. ACM Press.

[4] K. Bryson, M. Luck, M. Joy, and D. Jones. Applying agents to bioinformatics in geneweaver. In *Proc. of the Fourth Int. Workshop on Collaborative Information Agents*, Lect. Notes in Computer Science. Springer-Verlag, 2000.

[5] K. Decker, X. Zheng, and C. Schmidt. A multi-agent system for automated genomic annotation. In *Proc. of the Int. Conf. Autonomous Agents*, Montreal, 2001. ACM Press.

[6] K. S. Decker and K. Sycara. Intelligent adaptive information agents. *Journal of Intelligent Information Systems*, 9(3):239–260, 1997.

[7] S. S. Dritz, M. M. Chengappa, J. L. Nelssen, M. D. Tokach, R. D. Goodband, J. C. Nietfeld, and J. J. Staats. Growth and microbial flora of nonmedicated segregated, early weaned pigs from a commercial swine operation. *J. Am. Vet. Med. Assoc.*, 208:711–715, 1996.

[8] M. J. Freeman, C. H. Armstrong, L. Freeman-Sands, and L. Lopez-Osuna. Serological cross–reactivity of porcine reference antisera to *Mycoplasma hyopneumoniae*, *M. flocculare*, *M. hyorhinis* and *M. hyosynoviae* indicated by the enzyme–linked immunosorbent assay, complement fixation and indirect hemagglutination tests. *Can. J. Comp. Med.*, 48:202–207, 1984.

[9] R. F. Goodwin. Apparent reinfection of enzootic–pneumonia–free pig herds: early signs and incubation period. *Vet. Res.*, 115:320–324, 1984.

[10] R. F. Goodwin, A. P. Pomeroy, and P. Whittlestone. Production of enzootic pneumonia in pigs with mycoplasma. *Vet. Rec.*, 77:1247–1249, 1965.

[11] R. D. King, S. Muggleton, R. A. Lewis, and M. J. Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure–activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Natl. Acad. Sci*, 89:11322–26, 1992.

[12] R. D. King and M. J. Sternberg. Machine learning approach for the prediction of protein secondary structure. *J. Mol. Biol.*, 216:441–57, 1990.

[13] M. Kobisch and N. F. Friis. Swine mycoplasmoses. *Rev. Sci. Tech. Off. Int. Epizoot.*, 15:1569–1606, 1996.

[14] E. Kretschmann, W. Fleischmann, and R. Apweiler. Automatic rule generation for protein annotation with the c4.5 data mining algorithm applied on swiss–prot. *Bioinformatics*, 17:920–926, 2001.

[15] D. Maes, M. Verdonck, H. Deluyker, and A. de Kruif. Enzootic pneumonia in pigs. *Vet. Quart.*, 18:104–109, 1996.

[16] C. J. Mare and W. P. Switzer. Mycoplasma hyopneumoniae, a causative agent of virus pig pneumonia. *Vet. Med.*, 60:841–845, 1965.

[17] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[18] S. Razin, D. Yogev, and Y. Naot. Molecular biology and pathogenicity of mycoplasmas. *Microbiol Mol Biol Rev*, 62:1094–1156, 1998.

[19] R. F. Ross. *Mycoplasmal diseases*. 7th edition, 1992.

[20] L. F. Schroeder and A. L. C. Bazzan. A multi-agent system to facilitate knowledge discovery: an application to bioinformatics. In *Proc. of the Workshop on Bioinformatics and Multi-Agent Systems (BIXMAS'2002)*, pages 44–50, Bologna, Italy, 2002.

[21] L. F. Schroeder, A. L. C. Bazzan, J. Valiati, P. M. Engel, and S. Ceroni. A comparison between symbolic and non–symbolic machine learning techniques in automated annotation the “keywords” field of swiss-prot. In A. L. C. Bazzan and A. de Carvalho, editors, *I Brazilian Workshop on Bioinformatics*, pages 80–87, Porto Alegre, RS, 2002. Soc. Bras. de Computação (SBC).

[22] A. Zaha. Projeto rede sul de anlise de genomas e biologia estrutural, 2001. In Portuguese.