A comparison between symbolic and non–symbolic machine learning techniques in automated annotation of the ”Keywords” field of SWISS–PROT

Luciana F. Schroeder[1], Ana L. C. Bazzan[1], João Valiati[1], Paulo M. Engel[1], and Sérgio Ceroni[2]

[1] Instituto de Informática, UFRGS
Caixa Postal 15064
91501–970 – Porto Alegre, Brazil,
{luciana,bazzan,jvaliati,engel}@inf.ufrgs.br
[2] Centro de Biotecnologia and Fac. de Veterinária, Univ. Fed. do Rio Grande do Sul
ceroni@dna.cbiot.ufrgs.br

Abstract. The aim of this work is to carry out a comparison between symbolic and non–symbolic approaches regarding the task of automated annotation of the field called Keywords in SWISS–PROT. The non–symbolic technique employed was a feedforward artificial neural network (ANN), while the symbolic ones was CN2. Using an ANN trained with the well–known Backpropagation algorithm over previously annotated data from public databases like SWISS–PROT, a classifier was built up that maps attributes of a specific protein to keywords encountered in SWISS–PROT and TrEMBL databases. The symbolic counterpart, CN2, builds a specific classifier for each keyword. Regarding the non–symbolic approach, the resulted classifier is much more compact than the symbolic counterpart. However, the symbolic one had a slightly better performance and is also more “readable” to the end user. The performance of the obtained classifier was evaluated using data taken out SWISS–PROT (for training) and TrEMBL for validation.

1 Introduction

With the increase in submission of sequences to public databases, there is a clear need for tools to generate automatic annotation. Following previous work on automated annotation, we employ symbolic and non–symbolic machine learning techniques as a method to generate automated annotation of the field “Keywords”, an important one in the SWISS–PROT database. The aim of this procedure is threefold: to complete the annotation of keywords which is far from adequate; to acquire experience to be able to propose automatic annotation on other (more complex) fields of the SWISS–PROT database; and to compare symbolic and non–symbolic techniques in this domain.

To test our approach, we employ data related to the organisms of the family *Mycoplasmataceae*, because one of these organisms is the object of the PIGS project [10]. This organism, *Mycoplasma hyopneumoniae*, is a bacterium which colonizes the respiratory tract of swine and is the primary agent of enzootic pig pneumonia. It causes considerable economic losses through retarded growth, poor food conversion, and increased susceptibility of pigs to infection by other organisms. The disease is one of the most relevant occuring in pigs in southern Brazil.

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: in the next section we briefly refer to related work, as well as to our previous work. Then, Section 3 describes the data employed, the symbolic and the non symbolic methods. In Section 4 we compare the results achieved.

2 Previous Work

There has been an explosion of data, information and computational tools coming out from the diverse genome projects. NCBI and EBI alone report huge databases, mostly not clearly structured. Here, technologies originally developed with other means in mind can help because the motivation behind their usage is the same: 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. In [2, 4] prototypes are described aiming at facilitating the process of annotation. Both works are based on information gathering: search, filtering, integration, analysis, and presentation of the data to the user.

Machine learning techniques have been widely used in bioinformatics (e.g. [5, 6]. Automatic annotation and machine learning are combined in [7]. The latter work describes a machine learning based 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 [7].

Basically, the authors have developed a method to automate the process of annotation regarding those keywords in SWISS–PROT, which is based on the algorithm called C4.5 [8]. This algorithm works on training data (in this case, previously annotated keywords regarding proteins). Such data is, in this case, mainly taxonomy entries, INTERPRO classification, and PFAM and PROSITE patterns. Given these data (called attributes), C4.5 derives a classification for a target attribute (in this case the keyword).

Since dealing with the whole data in SWISS–PROT at once would be prohibitive, it was divided in protein groups according to INTERPRO classification. Then each group is submitted to an implementation of C4.5 contained in the software package Weka [1]. Rules are generated and a confidence factor for each rule is calculated. The quality of the rules is evaluated by calculating a confidence factor 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://golgi.ebi.ac.uk/Spearmint.

The *Keywords* field in the SWISS–PROT database 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 the SWISS–PROT. Unfortunately, regarding the

[1] http://www.cs.waikato.ac.nz/ ml/

family of *Mycoplasmataceae*, a high number of proteins in SWISS–PROT are 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 regarding the Keywords field (in the data we collected, 378 out of 1894 had no keyword at all, while 896 had no attributes).

Therefore, the good results achieved in [7] 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. These results are reported in [1]. Here we give just a brief introduction for the sake of clarity.

We have initiated by reproducing the approach described in [7]. Soon we realized that since we reduced our universe to the proteins related to *Mycoplasmataceae* we could do better by modifying their method. Indeed, this 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 a confidence factor (CF). This may be confusing for the user. For instance, when two or more rules have close CFs, but nonetheless recommend contrary annotation (i.e. one does recommend the annotation while another does not), how should the user decide?

The approach we use in [1] is similar to that reported in [7] but we consider all applicable IPR Acc’s as attributes at once. Of course, taxonomy is no attribute in our case since we are dealing with a single family, namely the *Mycoplasmataceae*.

3 Methods

3.1 Data

The data collection was done in February 2002 by means of the SRS web site (version 6 at www.srs.ebi.ac.uk). Basically, we have performed a query on the SWISS–PROT database in which the Organism field included *Mycoplasmataceae* but the Keyword field does not include the word “hypothetical” or “Complete proteome”. This was done to eliminate hypothetical proteins from the training set.

Also, we have created a view for the SWISS–PROT database which associates this database with the IPRmatches (through a personal communication with SRS maintainers, we found out that the association with the INTERPRO was not working properly so we have used IPRmatches instead, which provides the required data as well). This view included:

- for SWISS–PROT: AccNumber, keywords
- for IPRmatches: IPR AccNumber

The number of proteins related to the *Mycoplasmataceae* family was 722 (Feb. 2002) while there were around 393 IPR Acc’s. Around 84 keywords appeared in the data. The next step was the retrieving process, which is very easy in SRS. This has generated a table (fields are delimited by “;”) partially depicted in Figure 1.

```
P47695;Cell divisionATP−bindingTransmembraneHydrolaseMetalloproteaseZinc;IPR000642IPR003593IPR003959IPR003960;
P75120;Cell divisionATP−bindingTransmembraneHydrolaseMetalloproteaseZinc;P75120;IPR000642IPR003593IPR003959IPR003960;
P47631Q49371;HydrolaseAminopeptidaseManganese;IPR000819;
SWISSPROT:AMPA_MYCPN;P75206;HydrolaseAminopeptidaseManganese;IPR000819;
P47707;HydrolaseAminopeptidaseManganese;IPR000819;
                                        ...
```

Fig. 1. Data Extracted from SRS

3.2 Symbolic Approach

This subsection describes the symbolic approach which uses data generated as explained before as input for the CN2 algorithm[3]. CN2 is a rule inductor algorithm developed by Peter Clark. It constructs simple, comprehensible production rules in domains where noise may be present. The rules produced by CN2 assume the form ”if ¡condition¿ then class”. The CN2 algorithm consists of two main procedures: a search algorithm performing a beam search for good rule and a control algorithm for repeatedly executing the search. During the search procedure, a rule is constructed by searching for a condition that covers a large number of examples of an arbitrary class C and few of other classes. Having found a good condition, the algorithm removes those examples it covers from the training set and adds the rule ”if ¡condition¿ then predict C” to the end of the list. For the remaining set, a new rule is constructed, until no further complexes of sufficient quality are found.

A typical file is partially depicted in Figure 2. The firsts lines indicate how the attributes are mapped for the 722 proteins (in this case we have 393 lines). The last of these is the target attribute (keyword). Finally, there come the 722 lines. Each of these is formed by the presence or absence of IPR characteristic separated by space.

```
IPR000005: yes no;
IPR000023: yes no;
IPR000032: yes no;
IPR000037: yes no;
...........
IPR004821: yes no;
class: Zincfinger  no;
@
no  no  no  no  no  no  no  no  no ...  no  yes  ...  no;
                           ...
```

Fig. 2. Input to the CN2 Algorithm – Class Zincfinger

To save time, we have generated the CN2 rules only for the keywords which appear in valid lines of the test data set (i.e. those from TrEMBL). A valid line has to have at least a keyword and at least an attribute IPR. A rule is depicted in Figure 3. It is possible to compare the structure of these rules to the similar ones produced by the Spearmint tool at the web site given in Section 2. Once the rules are generated, we have proceeded to the evaluation of the quality of these rules. Of course we avoid performing the test on the data set which was used to generate the rules. The obvious candidate to test data set is 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 same way already explained regarding the extraction of data from the SWISS–PROT database (i.e. query, view and save procedures as explained). Many proteins do not have either a keyword or an attribute. Therefore, these lines were deleted.

```
IF    IPR001241 = no
  AND IPR002936 = yes
THEN  class = Zincfinger  [8 0]

IF    IPR000191 = yes
THEN  class = Zincfinger  [3 0]
```

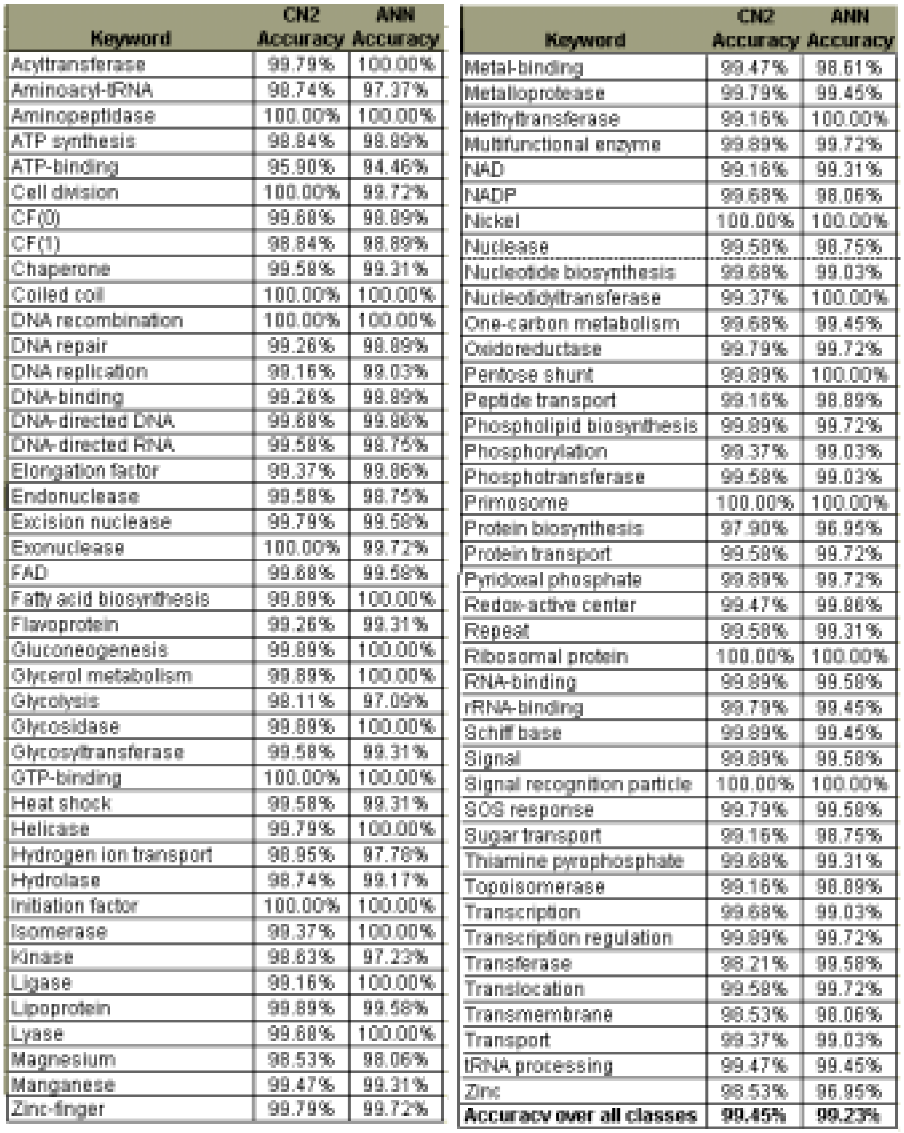
Fig. 3. Example of An Output from CN2 – Class Zincfinger

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 CN2) is checked. For instance, take the rule in Figure 3. It says that if the protein has the INTERPRO classification IPR002936 and does not have the IPR001241, then it should have the keyword "Zincfinger".

This procedure was repeated for each protein from the validation set (948 proteins from TrEMBL database). Figure 4 shows the accuracy per keyword gotten by each technique - CN2 and ANN. The accuracy estimation is calculate based on TP (True Positives, which means the number of examples correctly covered by the rule) plus TN (true negative, which means the number of examples correctly discarded by the rule) divided by the total number of instances from the validation set. The CN2 algorithm correctly predicted around 99% of the given keyword which is a very good result.

3.3 Non–Symbolic Approach

The data which was used was gathered the same way as for the symbolic tools. With the data sets defined, we have proceeded the training neural networking stage. The artificial neural network (ANN) model chosen was Multilayer Perceptron (MLP) with Resilient Backpropagation (Rprop) training algorithm [9]. This choice was done due to the philosophy used in the learning process and it effective and efficient training. The basic network architecture configuration was composed with 393 neurons in the input layer, 40 neurons in the hidden layer and 84 neurons in the output layer. It was stipulated



Keyword	CN2 Accuracy	ANN Accuracy
Acyltransferase	99.79%	100.00%
Aminoacyl-tRNA	98.74%	97.37%
Aminopeptidase	100.00%	100.00%
ATP synthesis	98.84%	98.89%
ATP-binding	95.90%	94.46%
Cell division	100.00%	99.72%
CF(0)	99.68%	98.89%
CF(1)	98.84%	98.89%
Chaperone	99.58%	99.31%
Coiled coil	100.00%	100.00%
DNA recombination	100.00%	100.00%
DNA repair	99.26%	98.89%
DNA replication	99.16%	99.03%
DNA-binding	99.26%	98.89%
DNA-directed DNA	99.68%	99.86%
DNA-directed RNA	99.58%	98.75%
Elongation factor	99.37%	99.86%
Endonuclease	99.58%	98.75%
Excision nuclease	99.79%	99.58%
Exonuclease	100.00%	99.72%
FAD	99.68%	99.58%
Fatty acid biosynthesis	99.89%	100.00%
Flavoprotein	99.26%	99.31%
Gluconeogenesis	99.89%	100.00%
Glycerol metabolism	99.89%	100.00%
Glycolysis	98.11%	97.09%
Glycosidase	99.89%	100.00%
Glycosyltransferase	99.58%	99.31%
GTP-binding	100.00%	100.00%
Heat shock	99.58%	99.31%
Helicase	99.79%	100.00%
Hydrogen ion transport	98.95%	97.78%
Hydrolase	98.74%	99.17%
Initiation factor	100.00%	100.00%
Isomerase	99.37%	100.00%
Kinase	98.63%	97.23%
Ligase	99.16%	100.00%
Lipoprotein	99.89%	99.58%
Lyase	99.68%	100.00%
Magnesium	98.53%	98.06%
Manganese	99.47%	99.31%
Zinc-finger	99.79%	99.72%
Metal-binding	99.47%	98.61%
Metalloprotease	99.79%	99.45%
Methyltransferase	99.16%	100.00%
Multifunctional enzyme	99.89%	99.72%
NAD	99.16%	99.31%
NADP	99.68%	98.06%
Nickel	100.00%	100.00%
Nuclease	99.58%	98.75%
Nucleotide biosynthesis	99.68%	99.03%
Nucleotidyltransferase	99.37%	100.00%
One-carbon metabolism	99.68%	99.45%
Oxidoreductase	99.79%	99.72%
Pentose shunt	99.89%	100.00%
Peptide transport	99.16%	98.89%
Phospholipid biosynthesis	99.89%	99.72%
Phosphorylation	99.37%	99.03%
Phosphotransferase	99.58%	99.03%
Primosome	100.00%	100.00%
Protein biosynthesis	97.90%	96.95%
Protein transport	99.58%	99.72%
Pyridoxal phosphate	99.89%	99.72%
Redox-active center	99.47%	99.86%
Repeat	99.58%	99.31%
Ribosomal protein	100.00%	100.00%
RNA-binding	99.89%	99.58%
rRNA-binding	99.79%	99.45%
Schiff base	99.89%	99.45%
Signal	99.89%	99.58%
Signal recognition particle	100.00%	100.00%
SOS response	99.79%	99.58%
Sugar transport	99.16%	98.75%
Thiamine pyrophosphate	99.68%	99.31%
Topoisomerase	99.16%	98.89%
Transcription	99.68%	99.03%
Transcription regulation	99.89%	99.72%
Transferase	98.21%	99.58%
Translocation	99.58%	99.72%
Transmembrane	98.53%	98.06%
Transport	99.37%	99.03%
tRNA processing	99.47%	99.45%
Zinc	98.53%	96.95%
Accuracy over all classes	**99.45%**	**99.23%**

Fig. 4. Accuracy for Each Class and Over All Classes

that the neural networks would be considered trained if they reached either the mean square error of $(1 * 10^{-3})$, or the maximum of 50 ephocs, or the gradient threshold of $(1 * 10^{-6})$. The structure and the parameters of the trained neural networks had been stored to be used in the validation stage, where samples of test were propagated and the outputs evaluated.

In a first stage, the set of 722 samples to training stage was used; the neural network was considered trained when the mean square error was reached (in 29 ephocs). In the validation of the trained neural model, a set with 948 samples was used.

In a second stage, we have used a training set of 948 samples; the network was considered trained when the maximum limit ephocs was reached; the mean square error stabilized in $2.6 * 10^{-3}$.

The trained ANN classifier is a single blackbox that maps each protein simultaneously onto the various keywords. In our experiment, 393 IPR Acc's were used as possible attributes for identifying a protein and 84 keywords were used as the domain for annotation.

The results are shown in Figure 4. In fact, we have considered the standard data of a contingency table: Acceptance Precision, Rejetc Precision, and Overall Accuracy as metrics of efficiency.

4 Comparison

The ANN is more compact and normally more efficient. Besides, it generates the rules after a single file containing all the data. However, as one of our goals is to integrate this methods in an environment for annotation of ORFs and proteins, it is important that the end user be able to “see” and analyse the rules generated. The biggest disadvantage of ANNs regarding this is the fact that the rules are not straightforward to the end user. One the other hand, the symbolic method was not able to cope with all entrances at once, that means, we had to generate rules one by one, i.e. one for each keyword. This is an important bottleneck in the process, of course. However, since the rules are supposed to be generated only once[2], this is not a significant shortcoming.

5 Conclusion

The main objective of this work was to carry out a comparison between symbolic and non–symbolic approaches to the task of automated annotation of the ”Keywords” field of SWISS–PROT. The comparison focused, on the one hand, on the accuracy of the generated model in predicting the correct keyword for previous unknown data. On the other hand, the compactness of the model was considered as an important element for comparison, once the symbolic approach requires the generation of a specific model for each keyword, while the non-symbolic approach generates just one model for fulfilling the task. Our results show the tradeoffs between both approaches. As the symbolic approach models each keyword separatedly, it can learn better each class. On the other

[2] In fact, if we consider the ever changing nature of the databases, we should speak of a periodic updating of the rule generation. However, this process can be done with low frequency.

hand, this leads to a considerable number of models, because in practical applications the number of keywords can reach a hundred or more. On the other hand, the neural network approach produces a very compact model, consisting in just one classifier with multiple outputs. However, the neural network model must consider all data in once, what leads to a slightly worse performace than the symbolic approach.

References

1. A. Bazzan, S. Ceroni, P. Engel, A. Pitinga, L. Schroeder, and F. A. Souto. Automatic annotation of keywords for proteins related to *Mycoplasmataceae* using machine learning techniques. In *European Conf. on Computational Biology (to appear)*, 2002.
2. 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.
3. T. Clark, P.and Niblett. The cn2 induction algorithm. *Machine Learning*, 3:261–283, 1989.
4. 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.
5. R. D. King and M. Sternberg. Machine learning approach for the prediction of protein secondary structure. *J. Mol. Biol.*, 216:441–57, 1990.
6. R. D. e. a. King. 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.
7. 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.
8. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
9. M. Riedmiller and H. Braun. Rprop – description and implementation details. Technical report, 1994.
10. A. Zaha and et al. Projeto rede sul de anlise de genomas e biologia estrutural, 2001. In Portuguese.