BIOINFORMATICS

Vol. 17 no. 10 2001
Pages 920–926

Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT

Ernst Kretschmann, Wolfgang Fleischmann and Rolf Apweiler

The EMBL Outstation, The European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Received on April 20, 2001; revised and accepted on July 8, 2001

ABSTRACT

Motivation: The gap between the amount of newly submitted protein data and reliable functional annotation in public databases is growing. Traditional manual annotation by literature curation and sequence analysis tools without the use of automated annotation systems is not able to keep up with the ever increasing quantity of data that is submitted. Automated supplements to manually curated databases such as TrEMBL or GenPept cover raw data but provide only limited annotation. To improve this situation automatic tools are needed that support manual annotation, automatically increase the amount of reliable information and help to detect inconsistencies in manually generated annotations.

Results: A standard data mining algorithm was successfully applied to gain knowledge about the Keyword annotation in SWISS-PROT. 11 306 rules were generated, which are provided in a database and can be applied to yet unannotated protein sequences and viewed using a web browser. They rely on the taxonomy of the organism, in which the protein was found and on signature matches of its sequence. The statistical evaluation of the generated rules by cross-validation suggests that by applying them on arbitrary proteins 33% of their keyword annotation can be generated with an error rate of 1.5%. The coverage rate of the keyword annotation can be increased to 60% by tolerating a higher error rate of 5%.

Availability: The results of the automatic data mining process can be browsed on http://golgi.ebi.ac.uk:8080/ Spearmint/ Source code is available upon request.

Contact: kretsch@ebi.ac.uk

TERMINOLOGY

This paper is about data, information, and knowledge on protein sequences. As far as we know there is no standard definition to distinguish between these concepts. In the following, we are going to use the definitions given below:

- *Data*: the measurable or observable facts, e.g. the sequence, the organism (in which the protein was found), the literature (in which it was mentioned), etc.
- *Information or annotation*: the statement of an aspect that is relevant or important to describe the protein as a whole or parts of it, e.g. ‘It is expressed in the mitochondrion’, ‘Amino acids 1–26 encode a Signal’, etc.
- *Knowledge*: the process that draws conclusions about an unknown protein using gathered information, e.g. ‘The protein sequence contains pattern x. Since all known sequences having this pattern belong to transmembrane proteins, this should also be a transmembrane protein.’
- *Data mining*: any technique that uses information to gain knowledge on data.

INTRODUCTION

How to obtain information about a protein? If the protein was biochemically characterized before and this information was entered into a database like SWISS-PROT, which is a completely human expert controlled and maintained database (Bairoch and Apweiler, 2000), one can simply make use of the provided information, i.e. a human being has used his knowledge to compose annotations on this very protein data and established a one to one relationship between this data and its annotation which can be used by others. However, often the information is incomplete, which is a fact for the majority of the known proteins. Many of those poorly annotated proteins are stored in databases like TrEMBL (Bairoch and Apweiler, 2000), which is only partly annotated by human experts but also by automated annotation systems like EDIT to TrEMBL (Möller *et al.*, 1999) and RuleBase (Fleischmann *et al.*, 1999; Apweiler, 2001). The protein can even be hypothetical, so there is no information available at all.

In these cases one mostly resorts to sequence similarity or signature searches, hoping to find well annotated protein features sharing some similarity with the protein in question. Apart from similarity searches against comprehensive, non-redundant protein sequence databases like SWISS-PROT and TrEMBL (Apweiler, 2000), the use of protein sequence signature databases such as Prosite (Hofmann *et al.*, 1999), PRINTS (Attwood *et al.*, 2000) or Pfam (Bateman *et al.*, 2000) can be helpful as are protein cluster databases like SYSTERS (Krause *et al.*, 1999) and CluSTr (Kriventseva *et al.*, 2001). In those cases, there is a many-to-many relationship between annotation and data, i.e. one annotation is stored for many proteins and one protein sequence might match various signatures and their annotation. Obviously, the process of gathering, analyzing, evaluating, and deriving information is time-consuming and cumbersome. It can be regarded as manual data mining across various databases.

We have developed a method to automate this process for a subset of the information available in SWISS-PROT, the Keyword Line. Keywords are particularly useful for analysis because they are controlled, limited in number (at the time of this writing there were 850 different Keywords allowed), they show little inherent structure or dependencies and are either annotated or not. These facts make automated knowledge acquisition much easier as for comment lines and description lines, which often are in unstructured free text.

The implementation uses the C4.5 data mining algorithm to detect decision trees which are an equivalent notation to rules. C4.5 shows particularly good results for non noisy data, which is the case for SWISS-PROT. The derived rules are not only fitting the training set, but are also human readable and kept short. This is obtained by an elaborated heuristic approach inherent to the standard algorithm. Also statistical evidence is given for every rule, which can be used to order rules in terms of confidence. This property can be used to select subsets of rules for different applications, i.e. only the highly confident ones for error critical purposes where coverage is less important and all of the generated rules where coverage is the main concern.

SYSTEM AND METHODS

Algorithm

One of the basic ideas of artificial intelligence algorithms is to derive knowledge from training sets and apply it on yet unknown data. The C4.5 algorithm expects input in a tabular format where the last column contains the target, in this particular case the information if a given keyword is present or not. The previous columns store core data about the proteins like taxonomy details or the presence of sequence patterns. The algorithm tries to derive the contents of the last column by using the information in the other columns.

To illustrate the procedure, a simple example for the SWISS-PROT proteins in InterPro (Apweiler *et al.*, 2000) IPR003009 is given. One of those proteins matches to Prosite pattern PS00487, three to Pfam pattern PF01493, five belong to mammalia and three have the Keyword ‘FAD’. The distribution is as follows.

	Prosite	Pfam		
	pattern	pattern	Mammalia	FAD
	PS00487	PF01493		
Q9ZNZ7	-	yes	-	yes
Q9JI00	-	-	yes	-
Q9ZL14	yes	-	-	-
Q9NYQ3	-	-	yes	-
Q9UJM8	-	-	yes	-
Q9NYQ2	-	-	yes	-
Q43155	-	yes	-	yes
Q9T0P4	-	yes	-	yes
Q9WU19	-	-	yes	-

Fig. 1. Example of data distribution in InterPro IPR003009 (only part of data is shown). The first column contains the SWISS-PROT accession numbers of some proteins in this entry.

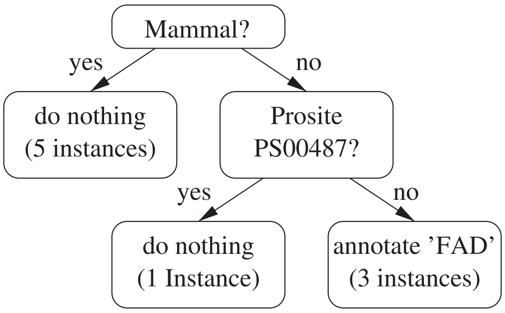

Fig. 2. Decision tree describing the data in Figure 1.

A decision tree is generated that has a preferably small number of leaves to make rules better readable and at the same time more reliable. Less leaves mean that on the average there are more examples per leaf that give the rule better statistical confirmation. In general, there are several possible equivalent decision trees. The example decision tree in Figure 2 covers all the instances in the training set in Figure 1.

But the decision tree in Figure 3 classifies the data more compactly. The problem of finding the optimal decision tree is known to be NP-complete (Hyafil and Rivest, 1976). C4.5 uses the gain ratio criterion, which is based on information theory and produces suboptimal trees heuristically (Quinlan, 1993). Note that if there are two instances having the same core data but a different annotation, there is no tree that classifies all examples of a training set correctly.

The precision of a tree can be checked by analyzing the number of correct and incorrect classifications it produces when applied on the training set. This analysis gives the number of True Positives (TPs) (annotation

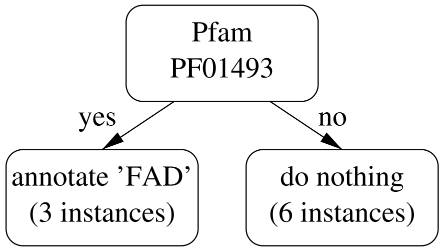

Fig. 3. Compact decision tree describing the data in Figure 1.

exists in instance and is predicted), True Negatives (TNs) (annotation does not exist in instance and is not predicted), False Positives (FPs) (annotation does not exist in instance but is predicted) and False Negatives (FNs) (annotation exists in instance but is not predicted).

A brute force implementation of the C4.5 algorithm could successively produce a decision tree for every allowed Keyword using all the protein core data available in SWISS-PROT. This procedure would produce huge data tables which can not be analyzed efficiently (every table would consist of more than 90 000 rows, one for each protein in SWISS-PROT). To produce decision trees with a satisfactory confidence fast enough, the number of instances for this application should be between 100 and 1000. Hence, a subdivision of SWISS-PROT into protein groups, which ideally contain similar proteins has to be performed. Thus, the grouping into proteins common to InterPro (Apweiler *et al.*, 2000) entries will be analyzed, since those entries usually contain a convenient number of similar proteins. Other groupings like using proteins common to CluSTr (Kriventseva *et al.*, 2001) entries were performed but not analyzed in detail. Brief investigations of some decision trees starting from CluSTr entries showed similar results to that starting from InterPro entries.

Extensions

The algorithm produces a large amount of rules with varying qualities. In many cases the annotation is not a result of sequence signature or taxonomy and therefore a decision tree trying to classify instances on this basis will produce annotation at random. The application of those trees on unknown data would lead to a massive error rate. Therefore, a selection of the more trustworthy rules has to be made and evaluated.

There are two steps of statistical evaluation of the results: firstly, not every generated rule is suitable to be applied, since many proved to have either a too high ratio of FPs to TPs or simply too few sample cases to derive a good statistical confirmation. Therefore, a smart criterion had to be used to select only the best rules with a reasonably high confidence. Secondly, once rules with a reliability over a given threshold are selected, an estimation has to be performed of how well they will perform on unknown data in terms of coverage and error rate. This aspect was tested by a tenfold cross-validation (see Results).

The standard algorithm was designed to classify instances into groups where there is an interest in all classes. Yet in this particular application there is no need for rules suggesting the non-annotation of certain Keywords. The standard statistical evaluation implemented in C4.5 was tried as a method to order rules in terms of quality. Parameters to trigger the calculation were adapted to the particular problem but the results were unsatisfactory. Therefore a procedure was chosen that derives confidence by exclusively using the number of TP and FP examples. The formula calculates the following value of likelihood: given the number of TP and FP examples it calculates, which rules lie above a given threshold in 95% of all cases. To illustrate the idea: suppose drawing from an urn containing an infinite number of balls. Drawing ten black balls and one white ball, over which value does the true ratio black to white balls in the urn lie in 95% of the cases? (TP = True Positives, FP = False Positives, c = confidence)

$$\begin{aligned}
z &= 1.96 \qquad \text{(constant for 95\%)}\\
n &= \mathrm{TP} + \mathrm{FP}\\
p &= \text{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}\\
c &= \text{confidence} = \frac{p + \frac{z^2}{2n} - z * \sqrt{\frac{p}{n} - \frac{p^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}.
\end{aligned}$$

Formula 1. Ordering rules in terms of confidence. The formula depends on TP and FP examples exclusively. Confidence gives the value above which all experiments would lie when an urn experiment was perfomed with the same distribution of correct and false outcomes.

Figure 4 gives a short overlook, for which confidence would be calculated for given numbers of TP and FP examples. To be introduced in the database, a rule had to have a confidence of 50% or more.

IMPLEMENTATION

The core application is Java based and uses the Weka Machine Learning Software package which is open source software and issued under the GNU General Public License (download at http://www.cs.waikato.ac.nz/~ml/weka/). The system is divided into a loader module that translates the core information stored in various databases into the tabular input format of the algorithm and an analyzer module that derives rules and stores the result into a newly created database to allow quick and easy access. Thus, the classical pipeline input–processing–output

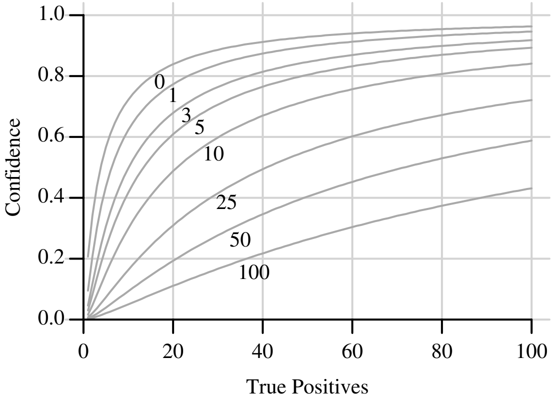

Fig. 4. Ratio TP to FP examples and the resulting confidence. The curves are for 0, 1, 3, 5, 10, 25, 50 and 100 FPs from top to bottom.

was implemented with the processing and output unit tied together, an approach that makes extensions and maintenance easier than that of a single monolithic application.

Two different modes of operation were implemented: One produces rules on the basis of all suitable SWISS-PROT proteins and writes them to a database. The other performs the cross-validation and evaluates rules without storing them. The data flow of both applications is shown in Figures 5 and 6.

A graphical user interface was developed that allows browsing of the generated information. It has some functionality implemented that can be valuable for the work of the professional annotators but also for a broader range of applications:

- It is possible to input the accession number of a protein in TrEMBL and get the suggested keyword annotation together with a confidence for each keyword.
- It is possible to track inconsistencies in SWISS-PROT by using SWISS-PROT both as a training and a test set. Sometimes it is not possible to find a rule without FNs and/or FPs. Those can be examined to validate their annotation.
- The manual generation of rules in RuleBase can be supported by proposing rules for a given set of proteins for manual processing.

RESULTS

Successively applied on the proteins assembled in each InterPro entry the algorithm generated 11 306 rules whose reliability was evaluated by a tenfold cross-validation. The quality of the rules depends on the quality of the data in the training set on one hand and on the bias between training data and the data on which rules are going to be applied on the other. The influence of the bias is difficult to measure and is further analyzed below.

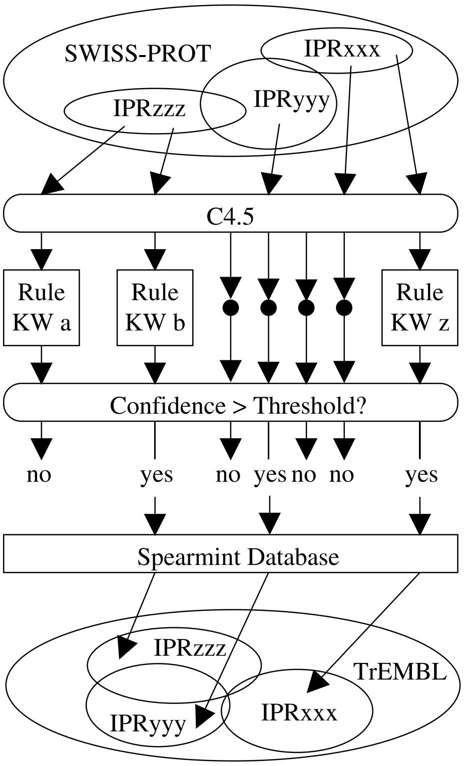

Fig. 5. Dataflow for a hypothetical production run. Rules are generated from InterPro families in SWISS-PROT, their confidence is tested against a given threshold (50%) and they are either discarded or added into the Spearmint database. From there they can be applied on proteins in TrEMBL. Note that rules starting from a given InterPro family are applied on the very same family and that the distribution of the InterPro families is different in SWISS-PROT and TrEMBL.

Within SWISS-PROT there are different levels of data quality due to a varying degree of experimental verification of different characteristics of a protein. Uncertain or predicted properties are categorized as probable, potential, putative and hypothetical with decreasing reliability in that order (Junker *et al.*, 1999; Apweiler, 2001; http://ch.expasy.org/cgi-bin/lists?annbioch.txt).

The general characterization status of a protein can be taken from the Description Line of the entry. The annotation of hypothetical proteins is usually bare and incomplete, which makes their usage in training sets unreasonable, hence they were not used for this purpose (apart from very few hypothetical proteins which are not marked as such in the Description Line). Probable and putative proteins were kept, but will be filtered out in future versions of the tool, since their annotation has unknown reliability.

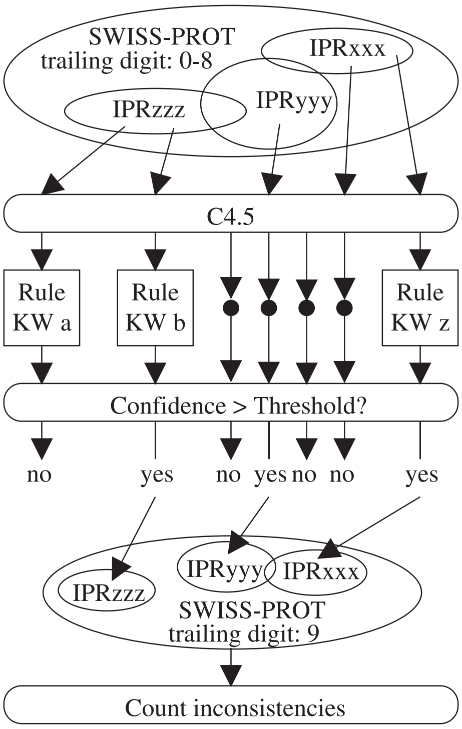

Fig. 6. Dataflow for a cross-validation run. Rules are generated from InterPro families in SWISS-PROT having trailing accession number digits 0–8, their confidence is tested against a given threshold (90 and 67%) and they are virtually applied on proteins in SWISS-PROT having trailing accession number digit 9.

Table 1. Fragments included in trainings set, confidence > 90%

End digit	No. of keywords	Covered keywords	No. of errors	% covered	% errors
0	28 225	9 629	214	33.36	2.22
1	28 033	9 533	214	33.24	2.24
2	27 899	9 579	155	33.78	1.62
3	28 040	9 553	172	33.46	1.80
4	27 498	9 340	214	33.19	2.29
5	28 058	9 609	223	33.45	2.32
6	28 247	9 647	171	33.55	1.77
7	28 049	9 386	210	32.71	2.24
8	27 748	9 380	192	33.11	2.05
9	28 129	9 414	206	32.73	2.19
	279 926	95 070	1971	33.26	2.07

Table 2. Fragments included in trainings set, confidence > 67%

End digit	No. of keywords	Covered keywords	No. of errors	% covered	% errors
0	28 225	17 456	1 049	58.13	6.01
1	28 033	17 337	966	58.40	5.57
2	27 899	17 148	1 001	57.88	5.84
3	28 040	17 348	907	58.63	5.23
4	27 498	17 037	1 045	58.16	6.13
5	28 058	17 311	1 142	57.63	6.60
6	28 247	17 457	983	58.32	5.63
7	28 049	17 274	1 087	57.71	6.29
8	27 748	16 905	998	57.33	5.90
9	28 129	17 279	948	58.06	5.49
	279 926	172 552	10 126	58.02	5.87

Protein fragments in the training set also reduce the data quality. Suppose having a number of proteins with a common sequence signature that induces a certain annotation. If there are only sequence fragments of those proteins contained in the database some might show the pattern, others might not depending on which part of the sequence is covered by the fragment. This is a highly random process introducing noise into the training set and their removal leads to lower error rates at the cross-validation. But it also leads to a bias between training data and target data, since the latter obviously will contain fragments as well as whole protein sequences.

Cross-validation has been performed on both, training sets including and excluding fragments. This was done by splitting the whole set of proteins contained in SWISS-PROT into ten parts of almost equal size. SWISS-PROT accession numbers always end with a digit. The digit does not encode any information about the protein as such and was therefore used as the split criterion. Nine parts of the split were used as training set to generate the decision trees, which were tested on the remaining tenth part called the test set. This procedure was repeated ten times, each time changing training and test sets. In each run the coverage and the error rate of the generated decision trees was measured. As stated above, not all rules are useful to be applied on unknown data. Rules can be selected using the confidence criterion described in Formula 1 to increase the reliability by decreasing the coverage and vice versa. Two tests were performed to test the value of that criterion and its influence on the observed error rate in the cross-validation: the first test used rules having a confidence of over 90% and the second test used rules with a confidence of over 67%. The numerical results are shown in Tables 1–4.

DISCUSSION

Reliability of the results

Obviously, the values of the observed error rate using the cross-validation give better results than the confidence obtained from Formula 1 would suggest. This is due to the very careful calculation of this criterion ($z = 1.96$) and could indicate a non-statistical distribution of SWISS-PROT proteins allowing the supposition of

Table 3. Fragments excluded in trainings set, confidence > 90%

End digit	No. of keywords	Covered keywords	No. of errors	% covered	% errors
0	25 471	8 443	121	32.67	1.43
1	25 303	8 381	120	32.65	1.43
2	25 275	8 597	94	32.64	1.09
3	25 418	8 480	102	32.96	1.20
4	24 824	8 286	138	32.82	1.67
5	25 338	8 542	155	33.10	1.81
6	25 676	8 600	101	33.10	1.17
7	25 310	8 277	146	32.13	1.76
8	25 158	8 335	123	32.64	1.48
9	25 590	8 348	136	32.09	1.63
	253 363	84 289	1236	32.78	1.47

Table 4. Fragments excluded in trainings set, confidence > 67%

End digit	No. of keywords	Covered keywords	No. of errors	% covered	% errors
0	25 471	15 632	803	58.22	5.14
1	25 303	15 578	752	58.59	4.83
2	25 275	15 482	762	58.24	4.92
3	25 418	15 679	668	59.06	4.26
4	24 824	15 334	808	58.52	5.27
5	25 338	15 560	907	57.83	5.83
6	25 676	15 773	769	58.44	4.88
7	25 310	15 529	845	58.02	5.44
8	25 158	15 285	775	57.68	5.07
9	25 590	15 671	748	58.32	4.77
	253 363	155 523	7837	58.29	5.04

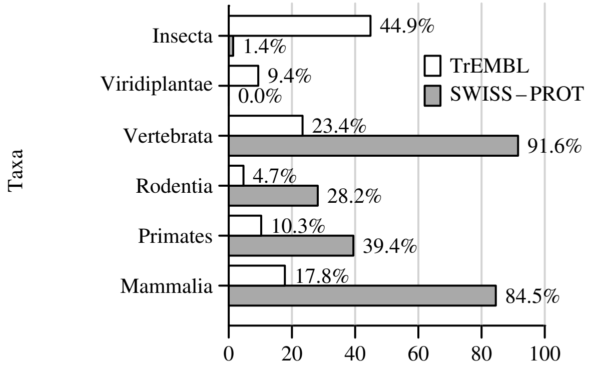

Fig. 7. Distribution of the Taxa from which the proteins in IPR000301 descend (selection).

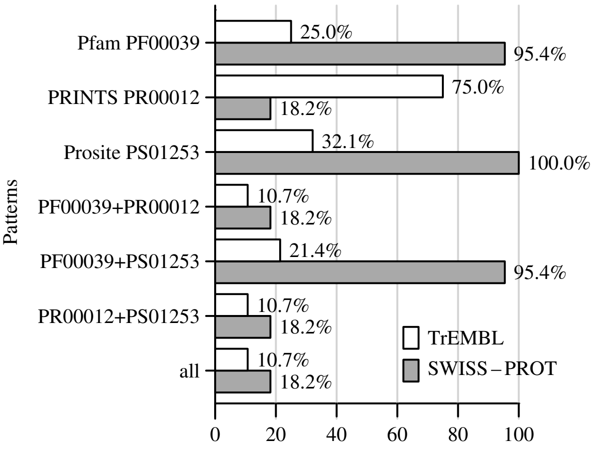

Fig. 8. Distribution of protein signatures in IPR000301 (selection).

higher confidences in rules than the statistics assuming a random distribution would suggest, e.g. finding four proteins matching to a common signature and sharing the same annotation without finding a counter-example produces on the average a rule with an error rate much less than the predicted 51% from Formula 1.

Furthermore, for the cross-validation it was assumed that the information in SWISS-PROT is true and without errors. Clearly, this precondition is not completely fulfilled, leading to an increased error rate for the cross-validation. In fact, there are three possible error sources that contribute to inconsistencies in the cross-validation:

(1) The rule suggests a Keyword which is not contained in a target due to a biological reason (True error).

(2) The Keyword has been forgotten to be annotated (Inconsistency in SWISS-PROT).

(3) The protein does not match the precondition of the rule due to a FP match to one of the signature databases (Inconsistency in SWISS-PROT).

Points 2 and 3 suggest that the real error rate is less than the one observed in the cross-validation.

The method assumes equal distribution of the proteins in the training set (SWISS-PROT proteins in an InterPro entry) and the data to be classified (TrEMBL entries or yet unknown sequences matching the same InterPro entry). This is clearly not the case as Figures 7 and 8 indicate.

This distribution is a result of the fact that TrEMBL proteins are not randomly chosen to be annotated and transferred to SWISS-PROT. For example, there is a high interest in human proteins, which leads to their over-representation in comparison to all other species.

Often whole protein families are annotated or updated. For some families, all proteins share the same signatures, which leads to an over-representation of these signatures in SWISS-PROT. Generating rules from these sets and applying them on proteins of different origin or matching different additional patterns might lead to systematic errors. An improved method of validation based on empirical evidence is under construction and is planned to be implemented in later versions of this tool.

Further developments

There are two ways for further developments. Firstly, the rule generation process can be improved and secondly, the rules can be applied in different applications. Currently, the following projects are under development:

- A web-based application that allows application of the rules not only on TrEMBL entries, but also on raw amino acid sequences.
- Mining for Description-, Comment-, and Feature Lines.
- Integration of other data mining techniques.
- Application of rules on entries in Ensembl (http://www.ensembl.org/) to predict functionality.
- Automated application of the rules on proteins in TrEMBL without human interaction.
- Automated rule generation on GO terms (The Gene Ontology Consortiuum, 2000) rather than on Keywords.

One of the most imperative tasks is to achieve an improved confidence calculation. The current routine does not use information about the number of TNs or FNs in the calculation and hence prefers the generation of frequent keywords rather than rare ones, i.e. general Keywords are produced more often than specific ones, but the latter are the more valuable ones. Extracting reliable rules for rare Keywords will certainly be a very useful improvement of the tool.

For all calculations independence between training and target set were assumed, which is clearly not the case. An empirical test to collect data about the influence of the bias is helpful to get a better picture about the performance of the rules on unknown data.

CONCLUSION

The presented method mines for Keyword annotation in SWISS-PROT using a Java implementation of the C4.5 algorithm on protein groups assembled in InterPro entries. The results are satisfactory in terms of coverage and confidence, yet it was pointed out that both aspects can be further improved. Including other methods to group proteins into sets containing similar proteins like CluSTr, Prodom (Corpet *et al.*, 2000) or others will help to increase the coverage while a refined statistical analysis will improve ordering of the generated rules in terms of reliability. This is supposed to lead to higher values of confidence.

At the current status, the rules can be used to support the manual annotation process performed by the SWISS-PROT database curators. It is also ready to be made publically available (http://golgi.ebi.ac.uk:8080/Spearmint/).

REFERENCES

Apweiler,R. (2000) Protein sequence databases. *Adv. Protein Chem.*, **54**, 31–71.

Apweiler,R. (2001) Functional information in SWISS-PROT: the basis for large-scale characterisation of protein sequences. *Briefings in Bioinformatics*, **2**, 9–18.

Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R, Durbin,R., Falquet,L., Fleischmann,W., Gouzy,J., Hermjakob,H., Hulo,N., Jonassen,I., Kahn,D., Kanapin,A., Karavidopoulou,Y., Lopez,R., Marx,B., Mulder,N.J., Oinn,T.M., Pagni,M., Servant,F., Sigrist,C.J.A. and Zdobnov,E. (2000) InterPro—an integrated documentation resource for protein families, domains and functional sites. *Bioinformatics*, **16**, 1145–1150.

Attwood,T.K., Croning,M.D.R., Flower,D.R., Lewis,A.P., Mabey,J.E., Scordis,P., Selley,J.N. and Wright,W. (2000) PRINTS-S: the database formery known as PRINTS. *Nucleic Acids Res.*, **28**, 225–227.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Bateman,A., Birney,E., Durbin,R., Eddy,S.R., Howe,K.L. and Sonnhammer,E.L.L. (2000) The Pfam protein falilies database. *Nucleic Acids Res.*, **28**, 263–266.

Corpet,F., Servant,F., Gouzy,J. and Kahn,D. (2000) ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. *Nucleic Acids Res.*, **28**, 267–269.

Fleischmann,W., Möller,S., Gateau,A. and Apweiler,R. (1999) A novel method for automatic functional annotation of proreins. *Bioinformatics*, **15**, 228–233.

The Gene Ontology Consortiuum (2000) Gene Ontology: tool for the unification of biology. *Nature Genet.*, **25**, 25–29.

Hofmann,K., Bucher,P., Falquet,L. and Bairoch,A. (1999) The PROSITE database, its status in 1999. *Nucleic Acids Res.*, **27**, 215–219.

Hyafil,L. and Rivest,R.L. (1976) Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett. 5*, **1**, 15–17.

Junker,V., Apweiler,R. and Bairoch,A. (1999) Representation of functional information in the SWISS-PROT data bank. *Bioinformatics*, **15**, 1066–1067.

Krause,A., Nicodème,E., Bornberg-Bauer,M., Rehmsmeier,M. and Vingron,M. (1999) WWW access to the SYSTERS protein sequence cluster set. *Bioinformatics*, **15**, 262–263.

Kriventseva,E.V., Fleischmann,W., Zdobnov,E.M. and Apweiler,R. (2001) CluSTr: a database of clusters of SWISS-PROT + TrEMBL proteins. *Nucleic Acids Res.*, **29**, 33–36.

Möller,S., Leser,U., Fleischmann,W. and Apweiler,R. (1999) EDITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation. *Bioinformatics*, **15**, 219–227.

Quinlan,,J.R. (1986) Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

Quinlan,J.R. (1993) C4.5*: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA.