Using the A3C System* for Annotation of Keywords – A Case Study

Cássia T. dos Santos**, Ana L. C. Bazzan***

Instituto de Informática, Universidade Federal do Rio Grande do Sul
Caixa Postal 15064, 90.501-970, Porto Alegre, RS, Brazil
{ctsantos,bazzan}@inf.ufrgs.br

Abstract. The A3C (Ambiente para Anotação Automática e Comparação de Genomas) system is an environment for agent-based annotation and comparison of genomes. This paper presents a case study involving the use of this system to annotate protein of the organism *Mycoplasma pneumoniae*. A3C allows the main tasks related to the annotation process to be accomplished in an integrated, flexible and automatic way, allowing the human expert to have more time to execute tasks that require their knowledge.

1 Introduction

Annotation of large numbers of protein sequences in the shortest time possible is one of the challenges of genome projects. The annotation means a characterization of a given protein structure, function, and its interactions with others proteins. Since it involves huge amount of data and is time-consuming, facilitating some parts of the process has been the motivation for the development of automated annotation environments (the reader is directed to [5] for a detailed descriptions of such tools).

The A3C (Ambiente para Anotação Automática e Comparação de Genomas) system is an environment for agent-based annotation and comparison of genomes. In the present paper, we focus on the annotation module. This is based on symbolic machine learning techniques and performs the annotation of some attributes regarding proteins. In particular, the field *Keywords* in the SWISS-PROT database is used in the annotation process. This field gives several hints to experts as to what regards proteins functions and structure. The several activities in the environment are distributed among agents which integrate multiple information resource and act simultaneosly.

This paper presents a case study involving the use of the A3C system to annotate protein from public database regarding the organism *Mycoplasma pneumoniae*. The paper is organized as follows. Section 2 presents the A3C system detailing its main tasks. Section 3 discusses the use of the system to annotate

* Project supported by CNPq (grant 550042).
** Author supported by CNPq.
*** Author supported by CNPq.

protein from *Mycoplasma pneumoniae* data. Finally, section 4 presents the final remarks.

2 The A3C System

The A3C aims to integrate the tasks related to the annotation (first level) and comparison of genomes (second level). The first level consists of tools to automate the annotation of proteins, identified after the sequencing. In the second level, algorithms compare the genomes to extract useful information to the first level. The objective is to discover the relations among the organisms, thus obtaining information about a given genome from the knowledge about others relevant ones already sequenced. In this work we focus on the annotation level.

The A3C annotation module consists of three layers: layer I aims at automating the tasks associated with the process of finding ORFs (Open Reading Frames); layer II is associated with the generation of rules to automatically annotate the keywords; and layer III allows the user to check the correctness of the automatic annotation. To this date, the main tasks on layers I and II are implemented, but not using fully technology of agents yet. The layer III is yet to be implemented. Below, the tasks on each layer are detailed.

Layer I. In the process of finding ORFs, the user provides a file containing a set of sequences (usually in Genbank[4] or FASTA[5] format). These sequences are submitted to a program to extract the ORFs and to output them to a database-like form. The ORFs are then used as input to the layer II.

Layer II. This is the kernel of the environment. Three main tasks are carried out: (a) data extraction and reformatting; (b) application of machine learning techniques to train data and produce the rules for annotation; and (c) the annotation regarding protein motifs and keywords. The first task deals with accessing the SWISS-PROT[6] [9] database to retrieve the training data. The retrieved data corresponds to the accession numbers for all attributes related to domains classification such as InterPro[7] [1], Prosite [10], Prodom [8], Pfam [4] and Prints [3] which appear in SWISS-PROT as cross-referenced databases, plus the corresponding keywords. After, the data is prepared in order to serve as entry for the machine learning tools.

The second task deals with generation of rules for annotation. Symbolic machine learning techniques are used, namely C4.5 [11] and CN2 [7]. These techniques usually have the same approach: a training set is given to the algorithm in order to generate the classifier (set of rules in C4.5 and CN2). The classifier will be then used to determine whether or not a given protein can be annotated with a given keyword.

4 http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html
5 http://www.ebi.ac.uk/fasta33
6 http://www.expasy.ch/sprot
7 http://www.ebi.ac.uk/interpro

It must be pointed out that a high number of proteins in SWISS-PROT is annotated as "hypothetical protein" regarding the keyword (20.826 protein according data obtained in August 2004). This keyword field does not add much value to the annotation. The keywords showing Gene Ontology (GO) [2] mapping are more expressive because GO is one of the most widely accepted ontologies in Bioinformatics. Having in mind that GO is an important source of structured knowledge of biological roles of proteins, the keywords on GO can be used to improve the annotation. In the A3C, the user can determine whether or not only keywords with GO mapping are considered in the annotation process.

The third task in layer II is the annotation regarding protein motifs. The annotation of motifs has two main steps: to read each ORF (output from layer I) and to query the databases of motifs. Each ORF is submitted to a process which consult the databases in order to get the list of motifs. Finally, having annotated each protein with the motifs, the rules generated by the machine learning algorithms are applied.

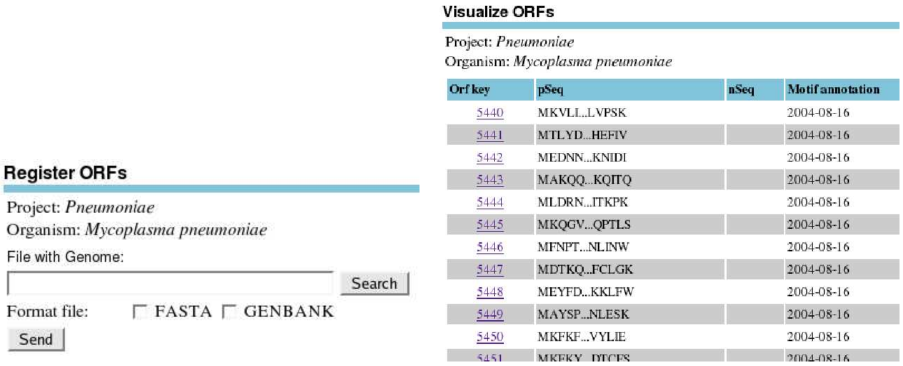
3 A Case Study Using A3C

Data from the public databases regarding the organism *Mycoplasma pneumoniae* was used. The user has to create a new Project (containing information about the target organism, coordinator and others details). After, s/he can access the main functionalities: register ORFs, annotate motifs, generate rules for annotation and annotate keywords.

Register ORFs. Initially, the user provides the system with a file containing the sequences of the genome (Figure 1(a)). This file is taken as input to a perl script, which extracts the list of ORFs and stores it in a Postgres database. After, information about the ORFs can be visualized (Figure 1(b)). The file used containing the ORFs was downloaded in June 2004 from NCBI[8]. 689 ORFs were found.

Annotation of motifs. The input to annotation of motifs is the list of the ORFs obtained in the previous step. Initially, each ORF is transformed to the FASTA format. After, they are submitted to the InterProScan in order to get the list of profiles. The InterProScan combines different protein signature recognition methods into one tool. It takes as input a given ORF and performs a data look-up from several databases of sequence motif (Interpro, Pfam, Prodom, etc.), comparing the sequence of aminoacids with the existing motifs. The output is the set of accession numbers for all attributes related to these databases. The ORFs are then annotated with the accession numbers found (relations in the A3C tables are created to associate each ORF to its motifs). Figure 2 shows an example of output from InterProScan, where the accession numbers to Interpro

[8] http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?val=U00089



(a) **(b)**

Fig. 1. (a) Register of ORFs; (b) visualization of ORFs.

(IPR001001) and Pfam (PF00712, PF02768) databases were found. The user can visualize the result of the annotation of motifs in Figure 3 (where the ORF was annotated with the motifs IPR001001, PF00712 and PF02768), accessing the details of the ORF from the interface of visualization (clicking on ORF key, as Figure 1(b)).

```
Sequence "SeqOrf_5440" crc64 checksum: AE3F5717962D3DFB length: 380 aa.

InterPro     IPR001001    DNA polymerase III, beta chain
method       AccNumber    shortName                              location
HMMPfam      PF00712      DNA_pol3_beta                          T[1-121] 1.9e-05 T[132-207] 1.7e-05
HMMPfam      PF02768      DNA_pol3_beta_3                        T[258-380] 1.1e-41
HMMSmart     SM00480      POL3Bc                                 T[17-377] 2.1e-129
HMMTigr      TIGR00663    dnan                                   T[1-380] 2.7e-32

InterPro     NULL         NULL
method       AccNumber    shortName                              location
Seg          seg          seg                                    ?[2-22]
superfamily  SSF55979     DNA clamp                              T[260-380] 5.7e-21 T[1-123] 5.9e-21 T[131-259] 4.2e-13
```

Fig. 2. Output from InterproScan.

Generation of rules for annotation. From the point of view of the end user, the generation of rules for annotation consists of two main steps: creating a criterion of search and generation of the rules. In the first step (Figure 4), the user informs the organism which will be used to search on SWISS-PROT and the minimal number of instances for each keyword (only the keywords satisfying this criterion are retrieved). In this search, all domain classifications are considered and the result is a set of statistical data about it (Figure 5). Local data from SWISS-PROT (as July 2004) was used. The minimum number of occurrences for each keyword was 50, in the present case. As a result, 687 protein were found. 137

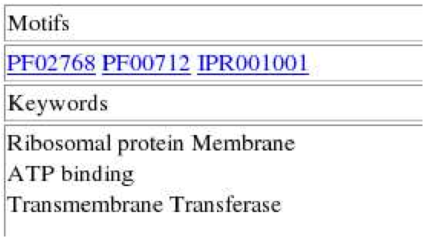


Fig. 3. Details of the ORF.

keywords appeared in the data, but only the keywords satisfying the threshold were considered. These keywords are: ATP-binding, Complete proteome, Hypothetical protein, Lipoprotein, Membrane, Palmitate, Ribosomal protein, Signal, Transferase, and Transmembrane. The exact number of attributes related to each database of sequence motifs appears in Table 1.

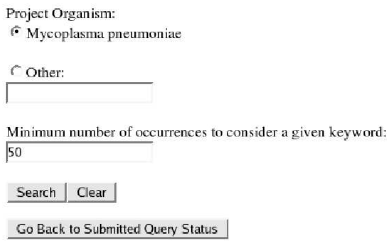


Fig. 4. Input data to search on SWISS-PROT.

Table 1. Number of attributes appearing in each database.

Data set	Number	Data set	Number
Intepro	675	Pfam	414
Prodom	105	Prosite	231
Prints	93	All	1518

The second step deals with the generation of rules for annotation (indicated as *Create rules* option in Figure 5). Given that the user can create several

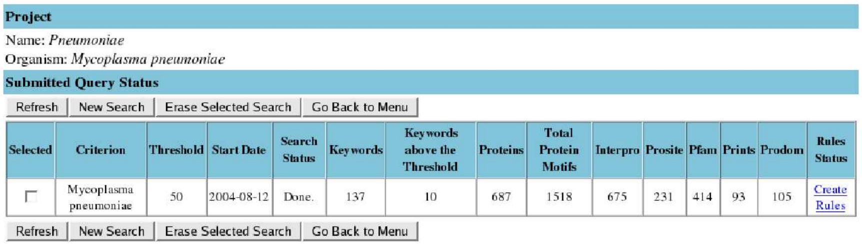


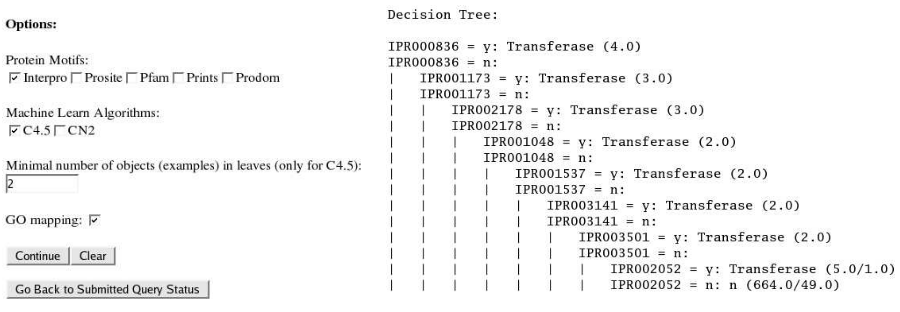
Fig. 5. Result of the search.

queries (varying the criterions of search), s/he must select the data set which will be used to generate the rules. In this step, the user indicates the set of motifs domains and the machine learning algorithm which will be used in the learning stage (Figure 6(a)). S/he also informs whether or not the rules will be generated considering only keywords which have a GO mapping. Moreover, the user can impose a constraint quality of the rules generated by C4.5, indicating the minimum number of instances that a rule must cover. These parameters are then used as input to perl scripts which prepare the training data to the corresponding machine learning tool, run the tool, and translate the output to the database-like form. We used the Interpro domain and the C4.5 algorithm to generate the rules, using only keywords with GO mapping[9]. In a preliminary study [6], we concluded that the most informative database is the Interpro, followed by Prosite and Pfam. Regarding keywords with GO mapping, only 5 of them have GO mapping namely: ATP-binding, Membrane, Ribosomal protein, Transferase and Transmembrane.

Figure 6(b) shows an example of output of the C4.5. In this specific case the rule refers to the keyword "Transferase". The "|" sign indicates an "AND". Thus, the rule basically suggests the annotation of the keyword "Transferase" for a given protein if it belongs to the IPR000836 family of proteins. If this is false, then, the rule goes on asking if it belongs to the family IPR001173. If the answer is yes, then that keyword should be annotated, otherwise continue reading the rule and performing the nested tests. At the end, either some "IF" part is satisfied, or the keyword should not be annotated.

Annotation of keywords. Once the rules were generated and the ORFs annotated with the motifs, the annotation of keywords is finished. Initially, the user must select the set of rules which will be used to annotate the proteins (Figure 7). After, a perl script performs three main tasks: reads each ORF and motifs from the databases; repeats the tests of the preconditions for each rule and concludes whether or not the keyword must be annotated; and creates entries

[9] List of keywords obtained from http://www.expasy.org/cgi-bin/keywlist.pl (July 2004).



(a) **(b)**

Fig. 6. (a) Input parameters to the generation of rules; (b) output from C4.5 algorithm.

on the database which associate each ORF to its keywords. Finally, the user can to visualize the result of the annotation (Figure 3).

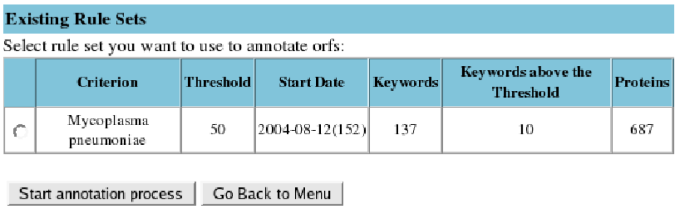


Fig. 7. Selection of rules to annotation process.

4 Final Remarks

The annotation module of the A3C system aims at automating the several tasks related to the annotation process of keywords and proteins. It is not a general purpose annotation tool. In order to demonstrate how to use it, this work presents a case study on annotation of proteins. The functionalities of the module were demostrated step-by-step using sequences of the organism *Mycoplasma Pneumoniae* as input data, generating annotation of keywords as output. The A3C permits that various input parameters be easily changed by the users, allowing the main tasks of the annotation process be performed in a more flexible way. Thus s/he can create several set of annotation, verifying the impact of the use of these parameters in the process, in order to select the most appropriated ones. Having in mind that these tasks are automated, the users have more time to execute tasks that really require their knowledge. The next activity is to address other fields of the SWISS-PROT database, possibly extending the use of the GO ontology.

References

1. R. Apweiler, T. K.Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M. D. R. Croning, R. Durbin, L. Falquet, W. Fleischmann, J. Gouzy, H. Hermjakob, N. Hulo, I. Jonassen, D. Kahn, A. Kanapin, Y. Karavidopoulou, R. Lopez, B. Marx, N. J. Mulder, T. M. Oinn, M. Pagni, F. Servant, C. J. A. Sigrist, and E. M. Zdobnov. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Research*, 29(1):37–40, 2001.
2. M. K. Ashburner, Ball C. A, J. A Blake, Botstein. D., H. Butler, J. M Cherry, A. P. Davis, K. Dolinsk, S. S. Dwight, J. T. Eppig, and et. al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
3. T. K. Attwood, M. D. Croning, D. R. Flower, A. P. Lewis, J. E. Mabey, P. Scordis, J. N. Selley, and W. Wright. Prints-S: The database formerly known as Prints. 28(1):225–227, 2000.
4. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall, and E. L. Sonnhammer. The Pfam protein families database. *Nucleic Acids Research*, 30(1):276–280, 2002.
5. A. L. C. Bazzan, R. Duarte, A. N. Pitinga, Schroeder L. F., S. C. Silva, and F. A. Souto. ATUCG–an agent-based environment for automatic annotation of genomes. *International Journal of Cooperative Information Systems*, 12(2):241–273, June 2003.
6. A. L. C. Bazzan and C. T. Santos. A study on the information provided by attributes taken from protein databases in the process of automating annotation of genome. *Internal Report*, June 2004.
7. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.
8. F. Corpet, J. Gouzy, and D. Kahn. Recent improvements of the Prodom database of protein domain families. 27(1):263–267, 1999.
9. Gasteiger E., Jung E., and Bairoch A. Swiss-prot: Connecting biological knowledge via a protein database. *Curr. Issues Mol. Biol.*, 3:47–55, 2001.
10. K. Hofmann, P. Bucher, L. Falquet, and A. Bairoch. The prosite database, its status in 1999. *Nucleic Acids Res*, 27:1, 1999.
11. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

This article was processed using the LaTeX macro package with LLNCS style