Reinforcement Learning Method for *BioAgents*

Célia Ghedini Ralha
Hugo Wruck Schneider
Maria Emília Machado Telles Walter
Universidade de Brasília, Depto. de Ciência da Computação
Caixa Postal 4466, CEP 70.919-970, Brasília, DF
Email: {ghedini,mia,hugowschneider}@cic.unb.br

Ana Lúcia Cetertich Bazzan
Universidade Federal do Rio Grande do Sul
Instituto de Informática
Caixa Postal 15064, CEP 91.501-970, Porto Alegre, RS
Email: bazzan@inf.ufrgs.br

***Abstract*— Machine Learning (ML) techniques are being employed in bioinformatics with increasing success. However, two problems are still prohibitive for symbolic ML methods: huge amount of data and lack of examples for training purposes. Thus, this paper introduces the use of reinforcement learning (RL), with the objective of dealing with these two drawbacks. Our work proposes and implement a RL method for the BioAgents system, in order to improve the annotation of biological sequences in genome sequencing projects. Experiments were done with real data from two different genome sequencing projects: *Paracoccidioides brasiliensis* - Pb fungus and *Paullinia cupana* - Guaraná plant. To assign reinforcement signals we have used reference genomes with curated annotations that are considered correct; these signals tackle specific databases and alignment algorithms. The results obtained with the inclusion of a RL layer in BioAgents were better compared with the system without the proposed method. Also, to the best of our knowledge, this is the first attempt to apply RL techniques to annotation in bioinformatics projects.**

I. Introduction

The Human Genome Project (HGP) is certainly one of the milestones of the 20^{th} century. HGP was finished in 2001 [35], [22], and, together with many other genome sequencing projects, have allowed great and fast development of techniques at both molecular biology and bioinformatics areas. Since then, great volume of biological information have been produced by molecular biology laboratories around the world, that have been stored in public databases [24], [13].

In recent years, new sequencing technologies [25], [30], that can produce billions of bases in a very short time have dramatically increased the amount of these data, and have presented new bioinformatics challenges [26]. Computer scientists need to develop new strategies to support the storage and the analysis of these data, using advanced computational techniques and methods.

Particularly, the annotation phase of a genome project has the objective of assigning biological functions to the DNA and RNA sequences generated by powerful short read sequencing machines. Besides *ab initio* gene finding programs to predict protein-coding genes, annotation techniques include finding similarities and producing alignments among the investigated sequences and sequences of related species, available in large public biological databases, like GenBank [13].

The annotation phase can be divided into two tasks: (i) automatic annotation, that uses computational programs to infer biological functions to each sequence; and (ii) manual annotation, in which the biologists guarantee accuracy and correctness to each sequence function, by using their knowledge to analyze and correct the function suggested by the automatic annotation. The former task is normally accomplished by approximate string matching algorithms (BLAST [6] and BLAT [20]) running on databases containing the sequences and their already identified functions, or methods to identify non-coding RNAs [7].

In the context of massively parallel DNA sequencing, manual annotation must be performed by computational tools. *BioAgents* is a multiagent system (MAS), originally designed to support manual annotation. Considering the current high-throughput DNA sequencing scenario, *BioAgents* can strongly support the annotation phase. The system simulates the biologists' knowledge and experience for annotating DNA and RNA sequences in genome sequencing projects. The MAS cooperative approach allows the interaction of specialized software agents in the reach of an objective [38], [37]. Specialized agents using different comparison algorithms, that interact with each other to suggest a function prediction to a DNA or a RNA sequence, may accomplish well the process of manual annotation. Previously, we have presented the architecture and a first prototype of *BioAgents* [23], [27].

This paper focuses on the use of ML techniques to bioinformatics. A symbolic ML approach with a RL method is presented and implemented in *BioAgents*. The main objective of this RL method is to deal with two main problems: huge amount of annotation data and lack of examples for training purposes, such as those used by a supervised method. Besides, we present experiments with two different genome sequencing projects: *Paracoccidioides brasiliensis* - Pb fungus and *Paullinia cupana* - Guaraná plant. In order to validate *BioAgents* suggestions, we have used two reference genomes: *Caenorhabditis elegans* and *Arabidopsis thaliana*, respectively, for Pb and Guaraná. The results obtained with the learning layer were better when compared to the system without the proposed method.

The rest of this work is divided as follows: in Section II, we discuss related work; in Section III, we present the *BioAgents* architecture, prototype, and the RL method; in Section IV, we describe and discuss the experiments; in Section V, we conclude and suggest future work.

978-0-7695-4210-2/10 $26.00 © 2010 IEEE
DOI 10.1109/SBRN.2010.27

II. Related Work

Bioinformatics is a research area concerned with the investigation of tools and techniques from computer science to solve problems from molecular biology (refer to [31] for more details). Several ML-based algorithms have been proposed in the literature for tasks such as finding particular signals associated with gene expression (e.g. [16], [10]) and others.

Protein function annotation includes classification of protein sequences according to their biological function. Ideally, manually curated databases are preferred over automated processing, in which case issues such as quality of data and compliance with standards could be carefully analyzed. However, this approach is expensive, if feasible. With the large number of protein sequences being daily included into databases, it is desirable to annotate new discovered sequences using automated methods. Approaches based on ML and data mining have been largely used, which explore functional annotation by using learning methods including decision trees and instance-based learning [11], [21], [18], neural networks and self-organizing maps [36], [32], and support vector machine (SVM) [14], [7].

Tetko et al. [34] proposed a method based on ML and the best score of alignments to annotate a sequence. In this work, the *5-fold cross-validation* technique was used with learning algorithms to predict the score of each protein pair. After the validation with different databases, the best score computed by the algorithms is chosen.

Rätsch et al. [28] proposed a learning system, named *mSplicer*, with the objective of improving the annotation of *C. elegans*, using ML techniques, SVM and label sequence learning. *mSplicer* correctly identified all the exons and introns of the *C. elegans* genome.

Agent-based technologies are useful in bioinformatics because the data is distributed among several sources, their contents are heterogeneous, and most of the work can be done in parallel, since the sources are independent. Hence, information agents can integrate multiple distributed heterogeneous information sources. There are only a few multiagent projects in the domain of bioinformatics. Next, some of these efforts are briefly reviewed.

Decker et al. [17] propose a prototype to automate the annotation of a virus sequence based on information gathering: searching, filtering, integrating, analysing, and presenting the data to the user. It uses the author framework DECAF, a MAS toolkit. The system has an overlapping multiagent organization: (i) basic sequence annotation, that aims to integrate remote gene sequence annotation from various sources with the gene sequences at the Local Knowledge Base Management Agent (LKBMA); (ii) query, that allows complex queries on the LKBMA via a Web interface; (iii) functional annotation, that is responsible for collecting information needed to guess the function of a gene using Gene Ontology (*GO*) [4].

Another MAS tool is the MASKS environment [29], that improves symbolic learning through knowledge exchange. The motivation is to mimic human interaction in order to reach better solutions. This tool supports a recent practice in data mining, that is the use of collaborative systems. Inductors are combined into a MAS with autonomy to improve individual models through knowledge sharing. These tasks are necessary because even if data mining is a powerful technique for knowledge extraction, none of the embedded algorithms are good in all domains.

Automated annotation and ML are combined in [21]. A ML approach was defined to automatically annotate genes based on rules using keywords of the *SwissProt* database [9]. This algorithm works on training data (previously annotated keywords regarding proteins) and generates rules to classify new instances of data. The training data comprises mainly taxonomy entries, motifs and patterns. Given these attributes, C4.5 derives a classification rule for a target class (keyword). Since dealing with the whole data in *SwissProt* at once would be prohibitive, the authors divided it in protein groups according to the InterPro classification.

III. *BioAgents* and the RL Method

As pointed out in Section I, *BioAgents* was originally developed to help biologists during the manual annotation phase of genome sequencing projects. This is a process in which the biologists' knowledge and experience are used to annotate genes. The system was first developed to simulate the manual annotation done by biologists, using the outputs produced in the automatic annotation phase, and interpreting these results according to the knowledge stored in *BioAgents*.

A. BioAgents: The Architecture

In this section we describe the new architecture of *BioAgents*, which is divided into four layers: interface, collaborative, learning and physical (Figure 1).

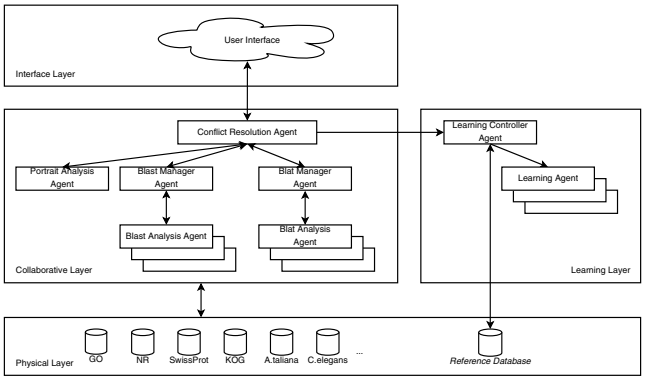


Fig. 1. *BioAgents* architecture.

This new version of *BioAgents* uses three comparison algorithms: *BLAST* [6], *BLAT* [20] and *Portrait* [7] to identify ncRNAs. Since databases used for annotation vary from project to project, we have used in our experiments specific databases (described in the *physical layer*) based on two projects developed in the Midwest region of Brazil: Pb fungus and Guaraná plant.

Normally, the RL methods are based on reinforcement signals, thus in this work we have used two reference genomes to derive these signals, *C. elegans* for Pb fungus and *A. thaliana* for Guaraná plant. Particularly, we have assigned points to the algorithm and database of each correct annotation suggested by *BioAgents*. Figure 1 presents the *BioAgents* architecture with the learning layer, detailed as follows:

- The *interface layer* is responsible for receiving the submitted requests (REQUEST message) and for returning the results for the users. The REQUEST message is passed to the conflict resolution agent at the collaborative layer and is composed by a list of sequences together with specific algorithms and databases to be analyzed by *BioAgents*. This layer is represented by a cloud, since the integration to other applications is possible by sending a REQUEST message to *BioAgents*.
- The *collaborative layer* is the architecture core, which is responsible for the suggestion of the annotation that will be returned to the *interface layer*, using the results produced by the execution of the algorithms over the databases of the *physical layer*. The *collaborative layer* is composed by the *conflict resolution agent* (CR), by the *manager agents* (MR), by the *analysis agents* (ANL) and by the *Portrait analysis agent* (POR).
  - The CR agent has the objective of submitting the requests received by the *interface layer* to the specialized MR agents. The CR agent creates a specific behavior working group to run the requested task, using a contract net protocol [38]. The REQUEST message is passed to the MR agents, according to their speciality, using the defined algorithm and database. After receiving the results of the MR agents, the CR agent decides which is the more appropriated suggestion to be sent to the *interface layer* using inference rules, and send a message to the *learning controller agent* (LC) to analyze the suggestion. In this work, we used *BLAST* and *BLAT* MR and ANL agents, but other algorithms can be easily integrated to *BioAgents*.
  - The MR agent receives a call for proposal (CFP)[1] message from the CR agent containing tasks, and accept the work according to its expertise (sending bids). The MR agent allocates a working group of ANL agents to perform the received tasks using the Achieve Simple Rational Effect (ASRE) protocol[2]. The MR agent then waits the suggestions of all the ANL agents, using previously defined production rules to consolidate these suggestions. As each MR agent is specialized on a particular tool (*BLAST* and *BLAT*), the MR agent is able to evaluate and to consolidate the results sent by its working group of ANL agents. The quantity of suggestions returned to the CR is based on the $Qtd(p)$ (Equation 1), where p is presented in Equation 2. This function was empirically defined using the experience of the biologists involved in the projects used in the experiments of this work. The idea is to limit the suggestions in projects with millions of sequences, where the number of good hits given by the algorithms are always limited and ordered by quality as shown in Equations 1 and 2.

$$Qtd(p) = \begin{cases} 5, & p=0 \\ \lfloor 5*\log(p) \rfloor + 5, & p>0 \end{cases} \quad (1)$$

  - Each ANL agent executes a particular algorithm (*BLAST* and *BLAT*) and uses a particular database. To analyze the results from the algorithm execution, inference rules are used (as presented at the end of Section III-B). The processed results with the correspondent hits are returned to the MR agent and saved at the RL database to be used by the learning layer agents. The quantity of suggestions returned to MR agent is also based on Equation 1.
  - The POR agent is activated when the CR agent does not receive any recommendation from the MR BLAST and BLAT agents for a specific request. The POR agent executes the non-coding RNA (ncRNA) *predictio* Portrait algorithm, which is based on SVM techniques, in order to verify if the sequence can be a ncRNA. *Portrait* was used since it is suitable for ncRNA analysis of transcriptomes from poorly characterized species.
- The *physical layer* is composed by the databases used in *BioAgents*. In all our case studies, we used the following data sources: *nr-GenBank* [5]; *GeneOntology* (GO) [4]; *Clusters of Orthologous Groups of proteins* (COG) [1] and the fungi databases of the nucleotide sequences of *Saccharomyces cereviseae* (SC) and *Schizosaccharomyces pombee* (SP). The databases needed to the comparisons must be integrated to the system.
- The *learning layer* is responsible for the RL method of *BioAgents*. It is composed by a LC agent and some *learning agents* (LE).
  - The LC agent is responsible for controlling the execution of the learning method. After receiving the message from the CR agent, the LC agent recovers the REQUEST message received by the CR agent and all the analysis done by the MR and ANL agents triggered by that request. LC agent executes BLAST with the reference genome to get the sequence *GO Accession Number* [8], [4]. A message and a *GO number* are passed to the LE agents.
  - A LE agent receives the message from the LC agent and execute the *BLAST* algorithm with the reference database: *Caenorhabditis elegans* for Pb project and *Arabidopsis thaliana* for Guaraná project. It uses the *GO Number* of each suggestion against the

[1]A CFP is part of a Contract Net (CN) interaction protocol defined by the IEEE Computer Society, the Foundation for Intelligent Physical Agents (FIPA) `http://www.fipa.org/`.

[2]The ASRE is a simple type of FIPA Request Interaction Protocol, which does not use AGREE and REFUSE messages between agents.

recommendations to compute the points (reward-r_t) related to the algorithm and to the database with the best recommended hit using *e-value* and score. The number of the LE agent is created by the LC agent according to the number of suggestions received. The LE agent results are saved in the reference RL database.

B. BioAgents: The Prototype

The *BioAgents* system was partially rewritten. In particular, a more efficient interaction protocol using CN was implemented between the CR and MR agents. Like the first version, it is implemented in *Java* language with *Java Agent DEvelopment Framework-JADE*, version 3.4.1. JADE [12] is FIPA compliant, allowing features needed to the asynchronicity of the agent communication language (ACL) messages and providing many interaction protocols, as the CN interaction protocol used in this new version of *BioAgents*. The *parsers* used by the ANL agents were implemented by adapting some libraries of the *BioJava* framework, version 1.4[3].

As in the first version, we have used an expert rule engine to allow agents reasoning, but at this new version we converted from JESS to Drools (JBoss Drools `http://www.jboss.org/drools`). With *Drools* we defined the biologists knowledge through the use of declarative rules according to the parameters defined on the related genome projects. In all experiments, we used the same production rules based on *BLAST* and *BLAT* results, following the biologists recommendations. *Drools* was specially developed to be integrated to *Java* language, which makes easier the development of *Java* applications. The defined rules have been elaborated to recognize and to predict DNA functions, which characterize the manual annotation task, where the biologists knowledge has to be formalized, in order to infer and suggest the annotations.

The rules used by the MR and ANL agents, with the *BLAST* and *BLAT* algorithms, capture the following biological knowledge:

- Check whether there are alignments having *e-value* less than or equal to 10^{-5} (value adopted by the biologists on both genome projects);
- Among the alignments satisfying the above restriction, select the lowest *e-value*;
- If the *e-values* are close, select the alignment with the highest *score*.

C. The RL Method

As mentioned before, ML techniques are being employed in bioinformatics with increasing success. Due to a lack of examples to train and test a symbolic ML approach, this paper introduces the use of RL to *BioAgents*. Our aim is to improve the annotation of biological sequences in genome sequencing projects, without necessarily having symbolic rules induced by for instance, a classifier. Also, the opportunity given by new sequencing machines, that produce a huge amount of data, points out in the direction of using RL methods, which typically need a long phase to learn, given that no examples are explicitly given.

According to [33], in RL problems an agent interacts with an environment to reach a goal. The agent perceives and interacts with the environment, using the state perceptions (s_t) and execute actions (a_t), as presented in Figure 2.

Note that agents and environment interact along a discrete period of time $t = 1, 2, 3, ...$ At each time step t, the agent perceives the state of the environment $s_t \in S$, and selects a possible action $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t.

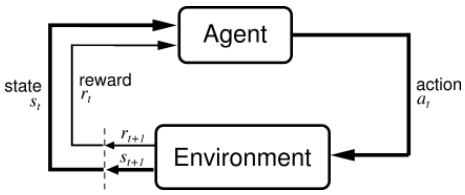


Fig. 2. Reinforcement Learning Model.

Every action $a_t \in A(s)$ has a reward $r_{t+1}(s_t, a_t) \in R$, where R is the set of possible rewards. After this interaction, the environment changes to state s_{t+1}. With this formalism, it is possible to compute the total reward value of the agent during its life cycle, starting from t as presented in Equation 2, where $\gamma \in (0, 1]$ is called a discount factor.

$$p = R(t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2}$$

It is important to mention that in the RL method employed here there is only one state (as e.g. in [15]). However, our method has two important RL characteristics: iterative learning and late reward. The iterative learning is implemented by the LE agents at the *learning layer*, which assign points to the algorithms and databases used for the suggested annotation. The reward is received only after the suggested annotation is checked. Assigned points are used only in the learning layer; thus the CR, MR, and ANL agents do not use them during their suggestions.

Regarding actions and rewards, we have set the reward for a_t as:

- 1 when one suggestion is considered correct and
- 0 when no suggestion is considered correct.

A suggestion is considered correct when both the *GO* Accession Numbers, respectively of the sequence of the reference genome (*C. elegans* or *A. thaliana*) used by the LE agent and of the sequence suggested by the CR agent, are the same.

IV. Experiments and Discussion

In order to validate the new version of *BioAgents*, we used data from two genome sequencing projects developed

[3]*BioJava* is a mature open-source project that provides a framework for processing biological data [19]. It is distributed under the Lesser GPL (LGPL) `http://www.biojava.org`.

at the Midwest Region of Brazil: Functional and Differential Genome from the *Paracoccidioides brasiliensis* (Pb) fungus [3] and Genome Project of *Paullinia cupana* Guaraná plant [2]. For the Genome Project Pb, the analyzed data was extracted from *BLAST* executed with *nr*, *COG* and *GO* databases; and from *FASTA* with *Saccharomyces cereviseae* and *Schizosaccharomyces pombee* fungi databases. For the Genome Project Guaraná, we used *BLAST* executed with *nr*, *KOG* and *SwissProt* databases.

To analyze the outputs from *BLAST* and *BLAT*, *BioAgents* used two parameters: the *expectation-value (e-value)* and *score* as presented in Section III-B. These parameters express the similarity between each sequence generated on the project with each sequence stored on the database. Both programs produce *alignments* between two sequences, which express their similarity by showing the correspondence among nucleotides or amino acids from one sequence relative to the other. The lower the *e-value*, the lower the error probability between the correspondences of both sequence of nucleotides or amino acids, and the higher the *score* the closer the sequences. Also, we have used *Portrait* for non-coding RNA analysis.

From the Genome Project Pb, $6,107$ sequences were analyzed (Table I). From these, $2,820$ genes were manually annotated, and $3,287$ were not. Note that $3,040$ annotations were suggested by *BioAgents* before the RL method, being $1,746$ correct when compared to the $2,820$ manual annotations, which corresponds to 61.91% of correct suggestions.

Observe that for the $3,287$ non-manually annotated genes, 533 were suggested by *BioAgents* and 447 were considered ncRNAs. With the RL method, *BioAgents* found even better results, being $3,119$ suggested annotations, with $1,910$ correct, which corresponds to 67.73% of correct suggestions. From the $3,287$ non-manually annotated genes, 566 were suggested, and the same 447 were considered as ncRNAs. We note that no tool was previously used in *BioAgents* to analyze ncRNAs.

TABLE I

RESULTS OF *BioAgents* APPLIED TO THE GENOME PROJECT PB.

	Without RL	With RL
Assembled ESTs	$6,107$	
Manually annotated genes	$2,820$	
Suggested genes	$3,040$	$3,119$
Correct suggested genes	$1,746$	$1,910$
Identified ncRNAs	447	447
Suggestions for genes not manually annotated	533	566

Considering the Genome Project Guaraná, $8,597$ sequences were analyzed (Table II). From these, $7,725$ genes were manually annotated and 872 were not. Note that $6,198$ annotations were suggested by *BioAgents* before the RL method, being $3,480$ correct when compared to the $7,725$ manual annotations, which corresponds to 45.04% of correct suggestions.

For the 872 non-manually annotated genes, 306 were suggested by *BioAgents* and $1,379$ were defined as ncRNAs. Considering the results after the RL method, we have found better results, being $6,276$ the number of suggested annotations with $3,601$ correct, which corresponds to 46.61% of correct suggestions. From the 872 non-manually annotated genes, *BioAgents* suggested 367 and considered $1,317$ as ncRNAs.

TABLE II

RESULTS OF *BioAgents* APPLIED TO THE GENOME PROJECT GUARANÁ.

	Without RL	With RL
Assembled ESTs	$8,597$	
Manually annotated genes	$7,725$	
Suggested genes	$6,198$	$6,276$
Correct suggested genes	$3,480$	$3,601$
Identified ncRNAs	$1,379$	$1,317$
Suggestions for genes not manually annotated	306	367

Figure 3 shows the results of Genome Project Pb and Genome Project Guaraná according to Tables I and II. Figure 4 shows the results of the Genome Project Pb and Guaraná in percentage. Based on the obtained results, we consider that *BioAgents* are helpful to biologists, especially in the context of massively parallel DNA sequencing.

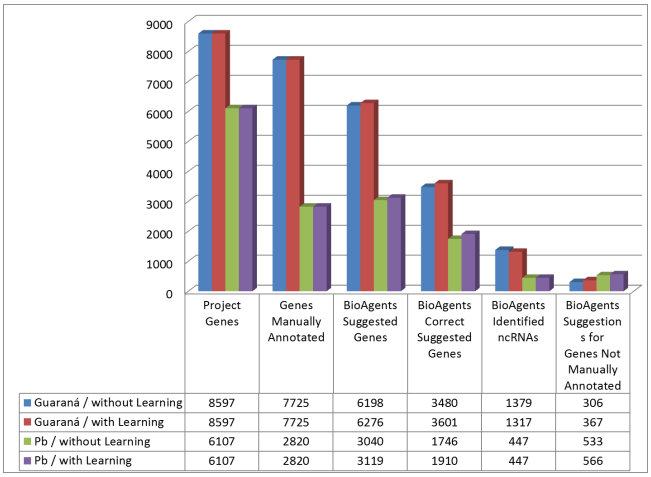
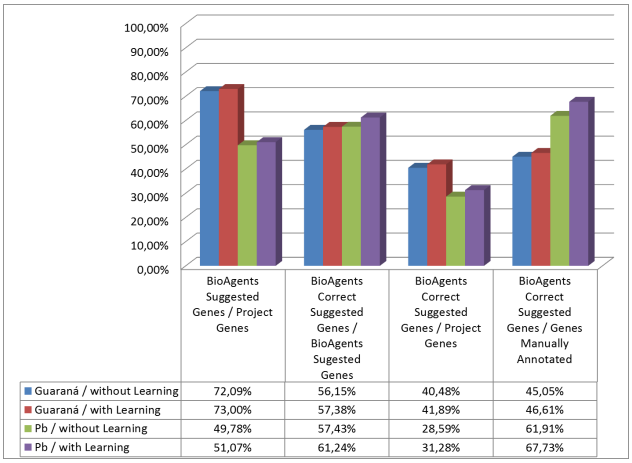


Fig. 3. Results of Genome Project Pb and Genome Project Guaraná.

V. CONCLUSIONS AND FUTURE WORK

In this article we presented a new version of *BioAgents*, a MAS to support annotation in genome projects, that includes a new protocol of interaction and a RL method. RL is interesting in this domain because the annotation process on genome sequencing projects is based on heterogeneous and dynamic environments. It uses different and distributed databases having large amounts of data continuously being modified. *BioAgents* has agents specialized on distinct tasks, so that they can act autonomously, using inference rules and a RL method to reward their correct actions.

Besides, we discussed the use of the new version of *BioAgents* in two different genome sequencing projects: *Paracoccidioides brasiliensis* (Pb) fungus and *Paullinia cupana* (guaraná) plant. The results increase from 61.91% to 67.73%, and $45,05\%$ to $46,61\%$ of correct suggestions, respectively on Genome Project Pb and Genome Project Guaraná. *BioAgents*



	BioAgents Suggested Genes / Project Genes	BioAgents Correct Suggested Genes / BioAgents Sugested Genes	BioAgents Correct Suggested Genes / Project Genes	BioAgents Correct Suggested Genes / Genes Manually Annotated
Guaraná / without Learning	72,09%	56,15%	40,48%	45,05%
Guaraná / with Learning	73,00%	57,38%	41,89%	46,61%
Pb / without Learning	49,78%	57,43%	28,59%	61,91%
Pb / with Learning	51,07%	61,24%	31,28%	67,73%

Fig. 4. Percent Results of Genome Project Pb and Genome Project Guaraná.

suggested 533 annotations before RL method and 566 with the RL method for the Project Pb; and 306 before RL method and 367 with the RL method for the Project Guaraná. These results were considered very promising by the biologists.

BioAgents may help to speed up the annotation process and to produce more refined annotations. We are currently working on an extended formulation of the RL problem, namely one that uses the Q-learning algorithm. In this direction, we plan to extend the presented approach to use a variety of data sets so that we can have more reward signals associated to defined actions. To the best of our knowledge, this is the first attempt to apply RL techniques to annotation in genome projects. The majority of the related work do not use RL approach to symbolic ML methods, so we might consider as future work to include Markov chain to improve the proposed RL method in *BioAgents*. We also plan to use *BioAgents* in high-throughput DNA sequencing projects.

References

[1] COG. `http://www.ncbi.nlm.nih.gov/COG/`.

[2] Genome Project Guaraná. `https://dna.biomol.unb.br/GR/`.

[3] Genome Project Pb. `https://dna.biomol.unb.br/Pb-eng/`.

[4] GO. `http://www.geneontology.org/`.

[5] nr-genbank. `http://www.ncbi.nlm.nih.gov/Genbank/`.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *J Mol Biol*, 215(3):403–410, 1990.

[7] R. T. Arrial, R. C. Togawa, and M. M. Brigido. Screening non-coding RNAs in transcriptomes from neglected species using Portrait: case study of the pathogenic fungus *paracoccidioides brasiliensis*. *BMC Bioinformatics*, 10(239), 2009.

[8] 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.

[9] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucleic Acids Res*, 27:49–54, 1997.

[10] P. Baldi and B. Søren. *Bioinformatics: the machine learning approach*. The MIT Press, Cambridge, MA, 1998. 351p.

[11] A. L. C. Bazzan, S. C. Silva, P. M. Engel, and L. F. Schroeder. Automatic annotation of keywords for proteins related to mycoplasmataceae using machine learning techniques. *Bioinformatics*, 18(S2):S1–S9, 2002.

[12] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[13] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucleic Acids Res*, 36(Database issue), 2008.

[14] C.Z. Cai, L.Y. Han, Z.L. Ji, and Y.Z. Chen. Enzyme family classification by Support Vector Machines. *Proteins: Structure, Function, and Bioinformatics*, 55(1):66–76, 2004.

[15] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752, 1998.

[16] M. Craven and J. W. Shavlik. Machine learning approaches to gene recognition. *IEEE Expert*, 9(2):2–10, 1994.

[17] K. Decker, X. Zheng, and C. Schmidt. A multi-agent system for automated genomic annotation. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 433–440, New York, NY, USA, 2001. ACM.

[18] M. des Jardins, P.D. Karp, M. Krummenacker, T.J. Lee, and C.A. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *Proc. International Conference on Intelligent Systems for Molecular Biology*, pages 92–99, 1997.

[19] R. C. Holland and et al. Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–7, 2008.

[20] W. J. Kent. BLAT: The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656–664, 2002.

[21] 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.

[22] E. S. Lander, L. M. Linton, B. Birren, and et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[23] R. S. Lima, C. G. Ralha, M. E. M. T. Walter, H. W. Schneider, A. G. F. Pereira, and M. M. Brigido. *BioAgents*: Um sistema multiagente para anotação manual em projetos de sequenciamento de genomas. In *ENIA'07: 6th Brazilian Meeting on Artificial Intelligence*, pages 1302–1310, Brazil, 2007.

[24] K. Liolios, N. Tavernarakis, P. Hugenholtz, and N. Kyrpides. The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide. *Nucleic Acids Research*, 34(Database-Issue):332–334, 2006.

[25] E. R. Mardis. Next-generation dna sequencing methods. *Annual Review of Genomics and Human Genetics*, 9(1):387–402, June 2008.

[26] M. Pop and S. L. Salzberg. Bioinformatics challenges of new sequencing technology. *Trends in genetics : TIG*, 24(3):142–149, March 2008.

[27] C. G. Ralha, H. W. Schneider, L. O. Fonseca, M. E. Walter, and M. M. Brígido. Using *BioAgents* for Supporting Manual Annotation on Genome Sequencing Projects. In *BSB'08: Proceedings of the 3rd Brazilian symposium on Bioinformatics-Lecture Notes in Bioinformatics*, volume 5676, pages 127–139, Berlin, Heidelberg, 2008. Springer-Verlag.

[28] G. Rätsch, S. Sonnenburg, J. Srinivasan, H. Witte, K.-R. Müller, R.-J. Sommer, and B. Schölkopf. Improving the *caenorhabditis elegans* genome annotation using machine learning. *PLoS Computational Biology*, 3(2, e20):0313–0322, 02 2007.

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

[30] S. C. Schuster. Next-generation sequencing transforms today's biology. *Nature Methods*, 5(1):16–18, December 2007.

[31] J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Co, 1997.

[32] T. C. Silva, P. A. Berger, R. T. Arrial, R. C. Togawa, M. M. Brigido, and M. E. M. T. Walter. SOM-PORTRAIT: Identifying non-coding RNAs using Self-Organizing Maps. In K. S. Guimarães, A. R. Panchenko, and T. M. Przytycka, editors, *BSB*, volume 5676 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 2009.

[33] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

[34] I. V. Tetko, I. V. Rodchenkov, M. C. Walter, T. Rattei, and H.-W. Mewes. Beyond the "best" match: machine learning annotation of protein sequences by integration of different sources of information. *Bioinformatics*, 24(5):621–628, 2008.

[35] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, and et al. The sequence of the human project. *Science*, 291(16):1304–1351, 2001.

[36] W.R. Weinert and H.S. Lopes. Neural networks for protein classification. *Applied Bioinformatics*, 3(1):41–48, 2004.

[37] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, USA, July 2000.

[38] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, 2nd edition, June 2009.