On Mining Protein Unfolding Simulation data With Inductive Logic Programming

Rui Camacho[1], Alexssander Alves[1], Cândida G. Silva[2], and Rui M. M. Brito[2]

[1] LIAAD & Faculdade de Engenharia, Universidade do Porto, Portugal
rcamacho, aalves@fe.up.pt

[2] Chemistry Department, Faculty of Science and Technology
and Center for Neuroscience and Cell Biology
University of Coimbra, Portugal
csilva@student.uc.pt, brito@ci.uc.pt

Summary. The detailed study of folding and unfolding events in proteins is becoming central to develop rational therapeutic strategies against maladies such as Alzheimer and Parkinson disease. A promising approach to study the unfolding processes of proteins is through computer simulations. However, these computer simulations generate huge amounts of data that require computational methods for their analysis.

In this paper we report on the use of Inductive Logic Programming (ILP) techniques to analyse the trajectories of protein unfolding simulations. The paper describes ongoing work on one of several problems of interest in the protein unfolding setting. The problem we address here is that of explaining what makes secondary structure elements to break down during the unfolding process. We tackle such problem collecting examples of contexts where secondary structures break and (automatically) constructing rules that may be used to suggest the explanations.

keywords: Inductive Logic Programming, Protein Unfolding

1 Introduction

In recent years, the identification of many human and animal diseases as protein misfolding disorders highlighted the importance of the protein folding problem, i.e. the process of conversion of a linear sequence of amino-acids into a functional tri-dimensional structure of a protein. After decades of efforts, this still is an unsolved problem in structural molecular biology. Central in health matters, it is today believed that protein unfolding events are responsible for triggering amyloidogenic processes in several proteins. These processes are at the origin of such disorders as Alzheimer, Parkinson, Bovine Spongiform

Encephalopathy (BSE), Familial Amyloid Polyneuropathy (FAP) and several other acquired and hereditary diseases. Thus, the detailed study of folding and unfolding events in proteins is not only important to the characterization of the mechanisms associated with several amyloid diseases but also is becoming central to the development of rational therapeutic strategies against these diseases. In this context, computer simulations based on molecular dynamics have been succesfully applied to explore and analyse the folding and unfolding events in proteins [5, 4, 1].

The opportunities that datamining methodologies offer to analyse, compare and contrast multiple protein unfolding simulations from different structural classes of proteins, and from amyloidogenic and non-amyloidogenic protein variants, opens new possibilities to allow the production of new knowledge or new views on the protein folding problem and its relationship with health and disease. Finding biologically significant rules may have important repercussions related to human and animal health, because a better understanding of the properties that make a protein amyloidogenic might help in the fight against this debilitating family of diseases - the amyloid diseases.

In order to find differences in the unfolding pathways of amyloidogenic (Am) and non-amyloidogenic(non-Am) variants of transthyretin (TTR), a protein associated with FAP, we use Inductive Logic Porgramming (ILP), a Multi-Relational Data Mining algorithm.

The main advantages of using ILP over competing technologies are sustained by the powerful expressive language to describe both data and the models. This powerful expressiveness as two major consequences: complex models may be constructed to "explain the data"; and the models are generally comprehensible, thus contributing to an insight on the phenomena that produced the data. Furthermore, ILP systems allow domain experts to provide almost any kind of information (ex., structured information like graphs) that may be helpful for the construction of the models. ILP systems may also combine in the same model symbolic relations with numerical computations.

The rest of the paper is organised as follows. Section 2 gives a brief introduction to Inductive Logic Programming. In Section 3 we describe the ongoing experimental work and preliminary results. Section 4 presents the conclusions on the preliminary work and points out future work.

2 ILP in a nutshell

Inductive Logic Programming (ILP) is a major field in Machine Learning with important applications in Multi-Relational Data Mining. The fundamental goal of a predictive ILP system is to construct models (usually called hypotheses) given background knowledge and observations (usually called examples in the ILP literature).

The task aim is to induce a logic program that given a set of positive and negative examples of the concept to learn, and some prior knowledge (or *background knowledge*), entails all positive examples and no negative example. In the context of this paper, positive examples correspond to events where protein secondary structure break, while the negative examples correspond to instants where there is no break on the secondary structure.

See [3] for an in-depth overview on ILP and [2] for a list of applications.

3 Preliminary Experiments

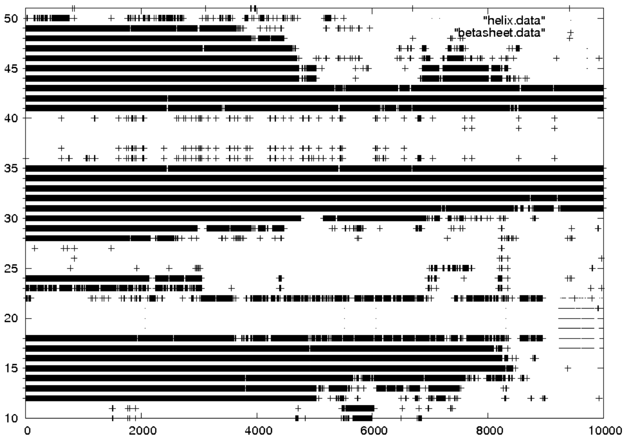

Fig. 1. Evolution of the secondary structure of WT-TTR during one unfolding simulation. The xx axis represents simulation time. The yy axis represents the position of a residue in the protein (only positions between 10 and 51 are represented). Thick lines indicate that the residue belongs to a beta sheet, thin lines indicate that the residue belong to a alpha-helix. We can see that (on top of the picture) the beta sheet between position 41 and 50 looses a substantial amount of residues near simulation time 5000. We can also see that a alpha-helix appears near simulation time 9000 between positions 17 and 23 (near the bottom right side of the picture).

The experiments we have designed have two objectives. First, to find rules that predict the circumstances in which secondary structures break down. Second, find rules that differentiate the break down process in the WT (wild type) and L55P protein variants. This later goal is very important since it may contribute to an explanation for the malignant behaviour of L55P-TTR.

We have used data from simulations of the protein transthyretin (TTR)[3]. The simulations [1] include 5 runs using the wilde type (WT) and 5 others using the amyloidogenic type (L55P). In each simulation run we collected, for each residue and instant of time, information concerning the secondary

[3] Reference 1TTA in the PDB (`http://www.rcsb.org/pdb/home/home.do`)

```
[Rule 1]  ssBreak(A,B,C,D,E) :-
              sasaSum(A,C,D,E,F), lteqSasa(F,40.0).
(‘‘A structure breaks if the sum of SASA of their residues is < 40’’)

[Rule 2]  ssBreak(A,B,C,D,E) :-
              sasaSumVariation(A,1,C,D,E,F), lteqDeltaSasa(F,-50).
(‘‘A structure breaks if the sum of SASA of their residues decreases more than
   or equal to 50 from one instant to the next in the simulation’’)

[Rule 3]  ssBreak(A,B,C,D,E) :-
              secStructure(D,sheet,C,F,G), gteqSize(G,10),
              sasaMinValue(A,C,D,E,H), gteqSasa(H,0.1).
(‘‘A beta sheet breaks if its size is greater than 10 residues and all its
   residues have a SAS greater than or equal to 0.1’’)
```

Fig. 2. Rules found by Aleph to predict the breakdown of secondary structure of proteins WT-TTR and L55P.

structure it belongs to and its Solvent Accessible Surface Area (SASA) value. Each SASA file has almost 7.3 MB and each secondary structure information has nearly 2.3MB. The total amount of data produced by the 10 simulations is nearly 100 MB. We have used the Aleph [6] ILP system.

From the original simulation data we constructed the ILP data set as follows. For each secondary structure we take its composition at instant 0 as a reference. Then we trace the simulation looking for an instant where a percentage (system parameter) of residues are no longer part of the structure. That instant marks a positive example. We also store *a window*[4] of simulation traces immediately preceding this event. This information is stored as ILP background knowledge and may be useful to explain the breaking of secondary structure. The simulation trajectories where there is no secondary structure break are also stored and a sample[5] of them is collected to construct the negative examples. With this filtering procedure we construct the positive and negative example's file and part of the background file (the one containing simulation information).

Apart from information concerning the simulation trajectories, the background knowledge includes a set of predicates useful to construct the models. So far we have encoded and used three major groups of predicates: predicates on the SASA value of residues; predicates on variation of SASA values and; general purpose relational predicates. In the first group we have predicates that compute the sum, the average, maximum value and minimum value of SASA of the residues in the structure. Predicates of the second group compute variations of the previous measures. The third group has predicates to compare numerical quantities.

[4] The size of the window is also a parameter for the filtering procedure.

[5] Another system parameter.

So far, we have found a small set of rules from which the most accurate are shown in Figure 2. These rules have good individual accuracy and are very easy to interpret. However they only cover ("explain") 53% of the positive examples – events where a secondary structure break. This value suggests that we need to improve the background knowledge, that is, we need more background predicates describing features necessary to "explain" the process.

4 Conclusions and Future Work

In this paper we have described an ILP-based approach to the automatic analysis of protein unfolding simulation data. We have addresses the specific problem of predicting the context where a protein secondary structure will break. Predictive rules were induced by the ILP system Aleph. The rules constructed so far are very easy to understand by the domain experts. On the other hand we have not yet been able to construct a set of rules that explain all the events where secondary structures break. This latter result suggest that further extensions to the background knowledge are required. We have also not yet found interesting rules that discriminate between the wilde type and the amyloidogenic variant of the protein.

Acknowledgments

This work has been partially supported by projects "Searching for high level rules in protein folding and unfolding: from amyloid diseases to protein structure prediction" (PTDC/BIA-PRO/72838/2006) and "ILP-Web-Service" (PTDC/EIA/70841/2006) and the doctoral fellowship SFRH/BD/16888/2004 (to CGS) by Fundação para a Ciência e Tecnologia.

References

1. R. M. M. Brito, W. Dubitzky, and J. R. Rodrigues. Protein folding and unfolding simulations: A new challenge for data mining. *OMICS: A Journal of Integrative Biology*, 8:153–166, 2004.
2. Saso Dzeroski. *Relational Data Mining.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
3. S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.
4. V.S. Pande, I. Baker, and Chapman J. et al. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68:91109, 2003.
5. J.E. Shea and C.L. Brooks. From folding theories to folding proteins: a review and assessment of simulation studies of protein folding and unfolding. *Annu. Rev. Phys. Chem.*, 52:499535, 2001.

6. Ashwin Srinivasan. The Aleph Manual, 2003. Available from http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph.