RNA secondary structure prediction by mapping Zuker's algorithm into Markov logic

Marco Lippi, Lukasz Popenda, Paolo Frasconi

Dipartimento Sistemi e Informatica,
Machine Learning and Neural Networks group,
Università degli Studi di Firenze

Abstract. RNA secondary structure prediction is a challenging task in computational biology. Several methods have been introduced to approach this kind of problem, and we believe that probabilistic inductive logic programming techniques like Markov logic networks may be the suitable framework to integrate multiple sources of information and a priori knowledge of the domain. Solving MAP inference within Markov logic is shown to be a natural transposition of Zuker's classic free energy minimization algorithm. This direction of research may also take advantage of structure learning techniques which directly learn rules from known RNA structures.

1 Introduction

Ribonucleic acid (RNA) is a biopolymer in which the monomers (nucleotides) are linked by phosphodiester bonds. RNA serves in a multitude of functions in living cells, such as catalysis, transport of proteins, regulation of transcription and translation [1, 2]. Determining the structure of an RNA molecule, given its sequence, is a crucial task towards understanding the activity of the cell. However, its accomplishment by experimental methods is very hard and in some cases even impossible, because of the limitations of the methods [3] and chemical properties of the RNA molecules. Thus, it is important to develop bioinformatics algorithms which could predict RNA structure, directly from sequence.

Predicting RNA structure using computational methods is a task which can be decomposed into different levels: at a first level, the *secondary structure* of the RNA molecule is predicted, by individuating its canonical base-pairs; a second more detailed step, which we do not cover within this work, would consist in the prediction of the *tertiary structure*, or three-dimensional shape of the molecule.

The prediction of RNA secondary structure is a well-studied problem, which has been addressed throughout the years using thermodynamics models [4, 5], comparative sequence analysis [6], free energy minimization [7, 8] and probabilistic context-free grammars [9, 10], which have recently been extended to tree adjoining grammars to handle the case of pseudoknots [11]. A classic approach for solving this problem has been introduced by Zuker [12, 13]. Zuker's method is based on free energy minimization, approximating the global energy of a structure with the sum of energies of the constituent motifs. The optimization problem

can be efficiently solved by a structural dynamic programming algorithm similar to the CKY algorithm commonly used for parsing natural language [14, 9]. For many RNA patterns, in fact, such as hairpins, internal loops, bulges and dinucleotide steps, sequence-dependent thermodynamics parameters have been experimentally determined and can therefore be used to compute the energy of larger fragments.

The approaches described above have been developed by several different communities: biologists, statistical physicists, mathematicians and computer scientists have followed parallel but in many cases overlapping directions of research, sometimes producing very similar algorithms. We argue that statistical relational learning [15], also known as probabilistic inductive logic programming [16] might be a suitable framework for the integration of such various approaches: in this setting, in fact, it is possible to combine in a single model information derived from various sources, like thermodynamics experiments, constraints determined from a priori knowledge of the domain, equations modeling internal and terminal loops. Moreover, while all these methods can only learn the *parameters* of their model, but cannot learn its *structure*, the use of a logic representation allows the employment of structure learning techniques, through which it is possible to learn clauses directly from the data describing known structures. In this paper, we employ Markov logic networks (MLN) [17], one of the most popular recently introduced methods for learning in relational domains, which combine first-order logic and probabilistic graphical models. Markov logic allows to describe objects and relations of some domain using a Markov network and a set of weighted first-order logic rules. In our case, the main idea is to exploit the log-linear structure of the joint distribution defined by the Markov random field associated with the MLN. This structure easily allows us to map potential energies derived from thermodynamics experiments into MLN weights. The high expressivity of first-order logic may also allow to introduce in the model several rules, able to handle complex energy functions which may not be simply used within Zuker's approach.

Some preliminary works trying to incorporate background knowledge and Constraint Handling Rules (CHR) within the process of RNA structure prediction have recently been proposed in [18, 19], yet without really exploiting the power of first-order logic formalism.

2 Zuker's algorithm

2.1 RNA motifs

In this Section we briefly revise the motifs used by Zuker to compute the overall energy of an RNA molecule.

Nearest neighbor base-pairs The first fundamental ingredient of Zuker's algorithm is given by the potentials of neighboring base-pairs (dinucleotide steps),

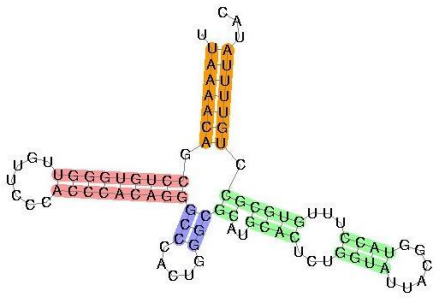

Fig. 1. Secondary structure of an enterovirus 5' cloverleaf cis-acting replication element [20]. This example contains a multi-branch structure consisting in four stem regions (depicted with colors), three hairpins (at the end of blue, pink and green stems), a 3×3 symmetric loop, a bulge within the green stem and a dangling end motif.

distinguishing between Watson-Crick base pairs (`A-U` and `G-C`) and wobble base-pairs (`G-U`). Such motifs can be represented by a set of quadruples $s_{nn} = \{(x_i, x_{i+1}, x_j, x_{j+1})\}$ such that x_i is bonded to x_{j+1} and x_{i+1} with x_j. As an example, the following configuration:

$$\begin{array}{lll} 5' & \texttt{CA} & 3' \\ 3' & \texttt{GU} & 5' \end{array} \tag{1}$$

brings a contribution of energy $\Delta G^{\circ}_{37} = -2.11\, kcal/mol$ [4].

Terminal loops (hairpins) The energy of terminal loops (sometimes referred to as hairpins) is derived from experimental data for stem loop stability by subtracting stabilities of stems calculated by the nearest neighbor model. The stability of hairpins longer than three nucleotides depends on the length of the loop, the stacking energies of the first mismatch and the closing base pair. Energy bonuses for special cases can also be included. The set of all possible admissible terminal loops $s_{hairpins}$ is described by the subsequences $x_i, \ldots, x_j$ of consecutive nucleotides, where x_i is partner with x_j, and none of the nucleotides in $x_{i+1}, \ldots, x_{j-1}$ is bonded.

Bulge loops A bulge is an interruption of the helical structure of RNA in one stem only. The energy of a bulge is determined by its length and its nearest neighbors. Bulges are formalized with a set of quintuples $s_{bulges} = \{(x_i, x_{i+1}, x_{i+2}, x_j, x_{j+1})\}$ such that (x_i, x_{j+1}) and (x_{i+2}, x_j) are base-pairs. For example, the configuration in the following example represents a bulge of length one:

$$\begin{array}{lll} 5' & \texttt{GXA} & 3' \\ 3' & \texttt{C U} & 5' \end{array} \tag{2}$$

which will produce different values of potentials, according to which is the nucleotide `X` in the loop.

Internal loops Internal loops are interruption of the helical structure of RNA in both stems. They can be symmetric or asymmetric, depending on the number of nucleotides which are present in each strand of the loop. For example, tandem mismatches are 2×2 internal symmetric loops, consisting in two opposing unpaired nucleotides in each strand, as represented in the following example:

$$\begin{array}{lll} 5' & \texttt{GXYA} & 3' \\ 3' & \texttt{CWZU} & 5' \end{array} \tag{3}$$

The energy of the configuration depends on the possible combinations of `X,W,Y,Z` nucleotides and on the nearest neighbors of the mismatch. The set of internal loops $s_{intloops}$ is a set of pairs of subsequences $(\{x_i, \ldots, x_{i+n}\}, \{x_j, \ldots, x_{j+m}\})$ such that (x_i, x_{j+m}) and (x_{i+n}, x_j) are base pairs, and none of the other nucleotides within the two subsequences is bonded.

Multi-branch loops The free energy of a multi-branch loop (also called junction) is computed as a function of the number of branches, the number of unpaired nucleotides, with an additional term which depends on the closing pairs.

2.2 Free energy minimization

Consider an RNA sequence $\mathcal{X} = x_1, \ldots, x_N$, with $x_i \in \{A, C, G, U\}$. The goal of Zuker's algorithm is to find the structure S_{opt} for which the free energy $\Delta G^\circ_{37}(\mathcal{S}_{opt})$ is minimum. The energy of a generic structure $\mathcal{S}$ is decomposed in the sum of contributions of the different motifs which are part of $\mathcal{S}$:

$$\Delta G^\circ_{37}(\mathcal{S}) = \sum_{s \in motifs(\mathcal{S})} \Delta G^\circ_{37}(s) \tag{4}$$

where $motifs(\mathcal{S})$ is the union of the sets of motifs described in previous section (s_{nn}, $s_{hairpins}$,...), which are admissible within structure $\mathcal{S}$. Going into more details, Equation 4 can be expanded by making more explicit the different motif categories. For simplicity of notation, we consider in the following only hairpins and internal loops:

$$\Delta G^\circ_{37}(\mathcal{S}) = \sum_{h \in hairpins(\mathcal{S})} \Delta G^\circ_{37}(h) + \sum_{i \in intloops(\mathcal{S})} \Delta G^\circ_{37}(i) \tag{5}$$

The minimum free energy configuration will therefore be structure S_{opt}, such that:

$$\begin{aligned} \mathcal{S}_{opt} &= \operatorname*{argmin}_{\mathcal{S}} \Delta G^\circ_{37}(\mathcal{S}) \\ &= \operatorname*{argmin}_{\mathcal{S}} \left(\sum_{h \in hairpins(\mathcal{S})} \Delta G^\circ_{37}(h) + \sum_{i \in intloops(\mathcal{S})} \Delta G^\circ_{37}(i) \right) \end{aligned} \tag{6}$$

In Section 3.2 we will explain how to implement Zuker's algorithm into Markov logic, showing the affinities between minimum free energy and MAP inference. Next Section will instead summarize the dynamic programming approach employed by Zuker.

2.3 Dynamic programming implementation

The minimization problem in Equation 4 can be efficiently solved using dynamic programming. Let $W_{i,j}$ be the minimum free energy of all admissible possible structures formed from a given subsequence $s_i, \ldots, s_j$, and let $V_{i,j}$ be the minimum free energy of all admissible structures formed from $s_i, \ldots, s_j$, with the additional constraint that nucleotides i and j are bonded. If i and j cannot be bonded, then $V_{i,j} = \infty$. Again, to simplify the notation, in the following we do not consider the case of multi-branch structures but we just take into account hairpins and internal loops (for full details, see [21, 22]). $W_{i,j}$ can therefore be recursively computed using the following:

$$\begin{aligned} W_{i,j} &= \min\{W_{i+1,j}, \min_{1<k\leq j}(V_{i,k} + W_{k+1,j})\} \\ V_{i,j} &= \min\{H(i,j), \min_{1<k<l<j}(V_{k,l}) + I(i,j;k,l)\} \end{aligned} \tag{7}$$

where $H(i,j)$ is the energy of a hairpin closed by pair (i,j), and $I(i,j;k,l)$ is the energy of an interior loop determined by base pairs (i,j) and (k,l). The minimum free energy structure is hence S_{opt} such that:

$$\Delta G^{\circ}_{37}(\mathcal{S}_{opt}) = W_{1,N}\,. \tag{8}$$

3 Markov logic implementation

3.1 Background

A Markov logic network (MLN) [17] consists in a set of first-order logic formulas $\mathcal{F} = \{F_1, \ldots, F_n\}$, and a set of real-valued weights $w = \{w_1, \ldots, w_n\}$, where weight w_j is associated to formula F_j. Together with a *finite* set of constants $C = \{c_1, \ldots c_k\}$ (corresponding to the objects of the domain), an MLN defines a Markov network where the set of nodes corresponds to all possible ground atoms, and there is an edge between two nodes if and only if the corresponding ground atoms appear together in at least one grounding of some formula F_j. While a first-order logic knowledge base can be seen as a set of *hard* constraints over possible worlds (if a world violates even only one formula, then it has zero probability), in Markov logic a world violating a formula will be *less probable*, but not impossible. Therefore, on the one hand, Markov logic networks extend first-order logic to handle uncertainty, by attaching weights to first-order logic rules; on the other, they can be seen as templates to build Markov networks, and hence they provide the full expressiveness of graphical models.

Maximum a posteriori (MAP) inference in Markov logic consists in finding the most probable world, according to the weights associated to the first-order logic formulas. In particular, since in many applications it is known a priori which atoms will be given as evidence($\mathcal{X}$), and which atoms will be queried ($\mathcal{Y}$), then MAP inference corresponds to the problem of finding the truth assignment of query atoms maximizing the sum of weights of satisfied clauses, given the evidence atoms. Any (weighted) satisfiability solver can be employed for this task: MaxWalkSAT [23], a weighted variant of WalkSAT local-search satisfiability solver, is one of the most used. MaxWalkSAT is a stochastic algorithm which, at each iteration, picks an unsatisfied clause at random and flips one of its atoms: with a certain probability p, the atom is chosen as the one maximizing the sum of satisfied clause weights when flipped; with probability $1-p$ it is chosen randomly. These stochastic moves help to escape local minima.

3.2 Energy minimization

The probability of a world x in Markov logic can be expressed by the following equation:

$$\begin{aligned} P(X=x) &= \frac{1}{Z}\exp\left(\sum_{i=1}^{|\mathcal{F}|} w_i n_i(x)\right) \\ &= \frac{1}{Z}\exp\left(\sum_{j=1}^{|\mathcal{G}|} w_j g_j(x)\right) \end{aligned} \tag{9}$$

where $n_i(x)$ is the number of true groundings of first-order formula F_i in world x, $\mathcal{G}$ is the set of *ground* clauses, and $g_j(x) = 1$ if and only if ground clause g_j is true within world x. The magnitude of a weight indicates how "strong" the corresponding rule is: the higher the weight, the less is the probability of a world violating that formula. Finding the most probable world means to find the world x such that $P(X = x)$ is maximum. Similarly, the prediction of the maximum probability structure for a given RNA sequence can be formalized in the following way:

$$P(\mathcal{S}=\hat{\mathcal{S}}) = \frac{1}{Z}\exp\left(\sum_{s\in motifs(\hat{\mathcal{S}})} \Delta G^{\circ}_{37}(s)\right) \tag{10}$$

If we now associate Boltzmann probability distribution to probability $P_{\hat{\mathcal{S}}} = P(\mathcal{S}=\hat{\mathcal{S}})$, we obtain:

$$P_{\hat{\mathcal{S}}} = \frac{1}{Z}\exp\left(-\frac{E_{\hat{\mathcal{S}}}}{kT}\right) \tag{11}$$

where $E_{\hat{\mathcal{S}}}$ is the energy of structure $\hat{\mathcal{S}}$, k is Boltzmann's constant, T the temperature and Z is the partition function.

From Equations 4, 10 and 11 it is straightforward to obtain a correspondence between Zuker's energy minimization problem and Markov logic MAP inference: just plugging the energy of a motif s as the opposite of the weight of the rule g_j which describe motif s:

$$w_{g_j} = -\Delta G^{\circ}_{37}(s) \tag{12}$$

the two problems become equivalent, in the sense that finding the structure with minimum energy corresponds to find the maximum probability world, if we map motifs into MLN clauses.

To make the same example as in Section 2.2, consider now RNA free energy only as a sum of contributions from hairpins and internal loops. If we model every hairpin h as a ground clause g_h and every internal loop i as a ground clause g_i, we can compute the minimum-energy structure in Markov logic as the world x_{opt} such that:

$$x_{opt} = \underset{x}{\operatorname{argmin}} \left(\sum_{g_h \in \hat{\mathcal{G}}_h(x)} w_{g_h} + \sum_{g_i \in \hat{\mathcal{G}}_i(x)} w_{g_i} \right) \tag{13}$$

which is equivalent to Equation 6, if we impose for each hairpin h that $w_{g_h} = -\Delta G^{\circ}_{37}(h)$, and the same for each internal loop i.

We map Zuker's equations into first-order logic clauses as described in the following. Note that MLN weights are simply the negative energies of the corresponding motifs.

Nearest neighbor rules Consider for example the two duplexes in example 1; a simple first-order logic rule which encodes such pattern is the following:

$$\begin{gathered} \texttt{Base(i,G)} \wedge \texttt{Base(i+1,A)} \wedge \texttt{Base(j,U)} \wedge \texttt{Base(j+1,U)} \wedge \\ \texttt{Bonded(i,j+1)} \wedge \texttt{Bonded(i+1,j)} \end{gathered} \tag{14}$$

to which we attach a weight $w = -\Delta G^{\circ}_{37} = +3.42(kcal/mol)$ [4].

Forbid non-canonical base pairs Rules like the following must have an infinite weight:

$$\begin{gathered} \texttt{Base(i,G)} \wedge \texttt{Base(j,A)} \Rightarrow \neg\texttt{Bonded(i,j)} \\ \texttt{Base(i,U)} \wedge \texttt{Base(j,C)} \Rightarrow \neg\texttt{Bonded(i,j)} \\ \ldots \end{gathered} \tag{15}$$

A nucleotide can have at most one partner base Also this rule must have an infinite weight.

$$\neg(\texttt{Bonded(i,j)} \wedge \texttt{Bonded(i,k)} \wedge \texttt{i!=j} \wedge \texttt{i!=k} \wedge \texttt{j!=k}). \tag{16}$$

1×1 symmetric loops A generic sequence-independent 1×1 symmetric loop can be encoded with the following rule:

$$\texttt{Bonded(i,j+2)} \wedge \texttt{Bonded(i+2,j)} \wedge \neg\texttt{Bonded(i+1,j+1)} \tag{17}$$

having a weight $w = -\Delta G^{\circ}_{37} = -0.40(kcal/mol)$.

Hairpin rules Several rules can be used to model hairpins. First of all, hairpins of length $L < 3$ can be forbidden using rules like the following:

$$\neg\texttt{Bonded(i,i+3)}. \tag{18}$$

A generic rule for a hairpin of length 4, for example, can be encoded by the clause:

$$\texttt{Bonded(i,i+5)} \wedge \texttt{InHairpin(i+1)} \wedge \texttt{InHairpin(i+4)} \tag{19}$$

to which we associate a potential $w = -\Delta G^{\circ}_{37} = -5.60(kcal/mol)$. Yet, for some particular sequences an energy bonus for the whole hairpin configuration is applicable, as in the following case:

$$\begin{aligned}\texttt{Base(i,C)} \wedge \texttt{Base(i+1,U)} \wedge \texttt{Base(i+2,A)} \wedge \texttt{Base(i+3,C)} \\ \wedge\texttt{Base(i+4,G)} \wedge \texttt{Base(i+5,G)} \wedge \texttt{Bonded(i,i+5)}\end{aligned} \tag{20}$$

which has a weight $w = -\Delta G^{\circ}_{37} = +2.80(kcal/mol)$.

4 Experiments

We present here some preliminary experiments, which have the only goal to show the fitness of the proposed approach. A data set of 120 RNA sequences was collected from PDB (Protein Data Bank), choosing RNA sequences with less than 50 nucleotides, and eliminating structures with non-canonical base pairs. Using Alchemy software [1] we built a Markov logic network with 72 clauses, and we compared results obtained by MAP inference with the ones produced by UNAFold software [8]. To compare the performance of the two systems, we measured the F_1 on base-pairing, as the harmonic mean between precision P (number of correct base pairs, out of predicted) and recall R (number of correct base pairs, out of true). The two systems produced very similar results: on 97 sequences out of 120 the predicted structure was identical; in 7 cases Markov logic achieved a better F_1, while in 16 cases UNAFold produced better results. The average F_1 produced by MLN and UNAFold was 94.0 and 95.8, respectively. These differences in the two predictors are due to small differences in the two models employed: in our MLN, for example, we still lack an efficient implementation of tandem mismatches (or 2×2 symmetric loops), which would otherwise produce a too large number of clauses. Also, we do not yet handle the case of loops and hairpins of arbitrary length, for which the use of Hybrid Markov logic networks [24] will be necessary. The use of such motifs could not be disabled for the computation of free energy in UNAFold, and for this reason the models of the two systems were not identical.

[1] http://alchemy.cs.washington.edu

5 Conclusions

We presented a Markov logic approach to RNA secondary structure prediction. The key idea is to map potential energies derived from thermodynamics experiments into MLN weights: in this way, MAP inference in the Markov logic network is shown to be equivalent to Zuker's free energy minimization algorithm. In simple preliminary experiments, the approach achieves similar results as Zuker's UNAFold prediction server. Further improvements of this method should arise from the use of structure learning techniques, through which it may be possible to learn rules directly from known RNA structures.

References

1. Lodish, H., Berk, A., Kaiser, C.A., Krieger, M., Scott, M.P., Bretscher, A., Ploegh, H., Matsudaira, P.: Molecular Cell Biology (Lodish, Molecular Cell Biology). 6th edn. W. H. Freeman (June 2007)
2. Cech, T.R., Atkins, J.F.: The Rna World (Cold Spring Harbor Monograph Series) (Cold Spring Harbor Monograph Series). Cold Spring Harbor Laboratory Press (October 2005)
3. Cantor, C., Schimmel, P.: Biophysical Chemistry Part II: Techniques For The Study Of Biological Structure And Function. 2nd edn. W.H. Freeman (2004)
4. Xia, T., Santalucia, J., Burkard, M.E., Kierzek, R., Schroeder, S.J., Jiao, X., Cox, C., Turner, D.H.: Thermodynamic parameters for an expanded nearest-neighbor model for formation of rna duplexes with watson-crick base pairs†. Biochemistry **37**(42) (October 1998) 14719–14735
5. Parisien, M., Major, F.: The mc-fold and mc-sym pipeline infers rna structure from sequence data. Nature **452**(7183) 51–55
6. Pace, N., Smith, D., Olsen, G., James, B.: Phylogenetic comparative analysis and the secondary structure of ribonuclease. Enzymology **180** (1989) 227–239
7. Do, C.B., Woods, D.A., Batzoglou, S.: Contrafold: Rna secondary structure prediction without physics-based models. Bioinformatics **22**(14) (July 2006)
8. Markham, N.R., Zuker, M.: Unafold: software for nucleic acid folding and hybridization. Volume 453. (2008) 3–31
9. Sakakibara, Y., Brown, M., Underwood, R.C., Mian, I.S., Haussler, D.: Stochastic context-free grammars for modeling rna. In: System Sciences, 1994. Vol.V: Biotechnology Computing, Proceedings of the Twenty-Seventh Hawaii International Conference on. Volume 5. (1994) 284–293
10. Knudsen, B., Hein, J.: Rna secondary structure prediction using stochastic context-free grammars and evolutionary history (1999)
11. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree adjoining grammars for rna structure prediction. Theoretical Computer Science **210**(2) (January 1999) 277–303
12. Zuker, M.: On finding all suboptimal foldings of an rna molecule. Science **244**(4900) (April 1989) 48–52
13. Mathews, D.H.: Revolutions in rna secondary structure prediction. Journal of Molecular Biology **359**(3) (June 2006) 526–532
14. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling. Volume 1. Prentice-Hall, Englewood Cliffs, NJ (1972)

15. Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
16. Raedt, L.D., Kersting, K.: Probabilistic inductive logic programming. In Raedt, L.D., Frasconi, P., Kersting, K., Muggleton, S., eds.: Probabilistic Inductive Logic Programming. Volume 4911 of Lecture Notes in Computer Science., Springer (2008) 1–27
17. Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M., Singla, P.: Markov logic. In Raedt, L.D., Frasconi, P., Kersting, K., Muggleton, S., eds.: Probabilistic Inductive Logic Programming. Springer, New York (2008) 92–117
18. Busch, A., Backofen, R.: Info-rna - a server for fast inverse rna folding satisfying sequence constraints. Nucleic Acids Research **35**(Web-Server-Issue) (2007) 310–313
19. Bavarian, M., Dahl, V.: Constraint based methods for biological sequence analysis. Journal of Universal Computer Science **12**(11) (2006) 1500–1520
20. Barton, D., O'Donnell, B., Flanegan, J.: 5' cloverleaf in poliovirus rna is a cis-acting replication element required for negative-strand synthesis. European Molecular Biology Organization Journal **20** (2001) 1439–1448
21. Zuker, M., Stiegler, P.: Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. Nucleic acids research **9**(1) (January 1981) 133–148
22. Mathews, D.H., Disney, M.D., Childs, J.L., Schroeder, S.J., Zuker, M., Turner, D.H.: Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of rna secondary structure. Proceedings of the National Academy of Sciences of the United States of America **101**(19) (May 2004) 7287–7292
23. Kautz, H., Selman, B., Jiang, Y.: A general stochastic approach to solving problems with hard and soft constraints (1996)
24. Wang, J., Domingos, P.: Hybrid markov logic networks. In: AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence, AAAI Press (2008) 1106–1111