# A Hybrid Symbolic-Statistical Approach to Modeling Metabolic Networks

Marenglen Biba, Stefano Ferilli, Nicola Di Mauro, and Teresa M.A. Basile

Department of Computer Science, University of Bari
Via E. Orabona, 4 - 70125 Bari, Italy
{biba,ferilli,ndm,basile}di.uniba.it

**Abstract.** Biological systems consist of many components and interactions between them. In Systems Biology the principal problem is modeling complex biological systems and reconstructing interactions between their building blocks. Symbolic machine learning approaches have the power to model structured domains and relations among objects. However biological domains require uncertainty handling due to their hidden complex nature. Statistical machine learning approaches have the potential to model uncertainty in a robust manner. In this paper we apply a hybrid symbolic-statistical framework to modeling metabolic pathways and show through experiments that complex phenomenon such as biochemical reactions in cell's metabolic networks can be modeled and simulated in the proposed framework.

**Keywords:** Systems biology, metabolic networks, symbolic-statistical frameworks, machine learning, statistical relational learning.

## 1 Introduction

Biological systems' behavior is determined by complex interactions between their building components. In Systems Biology [1] the main problem is to uncover and model how function and behavior of the biological machinery are implemented through these interactions. Since biological circuits are hard to model and simulate, many efforts [2] have been made to develop computational models that can handle their intrinsic complexity. In this paper we focus on a particular problem of Systems Biology that concerns the modeling of metabolic pathways. A metabolic pathway is a sequence of chemical reactions occurring within the cell. These reactions are catalyzed by enzymes which are particular proteins that convert metabolites (input molecules) in other molecules that represent the products of the reaction. These products can be stored in the cell under certain forms or can cause the initiation of another metabolic pathway. A metabolic network of a cell is formed by the metabolic pathways occurring in the cell.

Since a reaction can happen if the input molecules are available to the catalytic enzyme, a modeling framework must be able to model relations among entities. Symbolic approaches such as logic-based techniques have the potential to model relations in structural complex domains and there have been a growing number of

B. Apolloni et al. (Eds.): KES 2007/WIRN 2007, Part I, LNAI 4692, pp. 132–139, 2007.

biological applications of these methods [3]. First-order logic representations have also the advantage that models are easily comprehensible to humans. Moreover, since most part of biological systems performs its activity remaining hidden to the human modeler, machine learning techniques can play an important role in discovering latent phenomena. However, symbolic-only approaches suffer from the incapability of handling uncertainty. In models built with symbolic-only approaches, the learned rules are deterministic and do not incorporate any kind of mechanism for uncertainty modeling. On the other side, biological systems intrinsically behave in a uncertain fashion with many interactions probable to happen. Since cell's life is determined by the most probable interactions, handling uncertainty is crucial when the cell's machinery must be modeled. Statistical approaches based on the probability theory represent a valuable mechanism to govern uncertainty. However, observations of biological systems rarely reflect exactly what happens inside them. Therefore, estimation techniques are precious in order to model what we cannot observe. Statistical machine learning methods have the ability to learn probability distributions from observations and hence are suitable for modeling biological systems. On the other side, statistical-only approaches rarely are able to reason about relations and/or interactions among biological circuits as symbolic approaches do. Hence, there is strong motivation on developing and applying hybrid approaches to modeling biological systems.

The paper is organized as follows. Section 2 describes the problem of modeling the aromatic amino acid pathway of yeast and the necessity for symbolic-statistical machine learning. Section 3 describes the hybrid framework PRISM. Section 4 describes modeling of the problem in the framework PRISM. Section 5 presents experiments and Section 6 concludes and presents future work.

## 2 Metabolic Networks

Metabolic networks can be represented as graphs where nodes represent reactions and there are two kinds of arcs, those entering the node and labeled with the input molecules and those exiting the node and labeled with the products of the reaction. Fig. 1. shows part of the aromatic pathway for Yeast presented in [4]. The node represents the reactions made possible by the enzymes 2.5.1.19, 4.6.1.4 and 5.4.99.5 (KEGG identifiers are used for enzymes and reactions). A possible reaction is that of converting the metabolites C00074 and C00008 into the products C00009 and C01269. Then the metabolite C01269 takes part into another reaction made possible by the enzyme 4.6.1.4 and so on. Such interactions can be easily expressed using a first-order logic representation. For example the interactions can be expressed by the following predicates:

```
enzyme(2.5.1.19, reaction_2_5_1_19, [C00074,C00008], [C00009,C01269]).
enzyme(4.6.1.4, reaction_4_6_1_4, [C01269], [C00009,C00251]).
enzyme(5.4.99.5, reaction_5_4_99_5, [C00251], [C00254]).
```

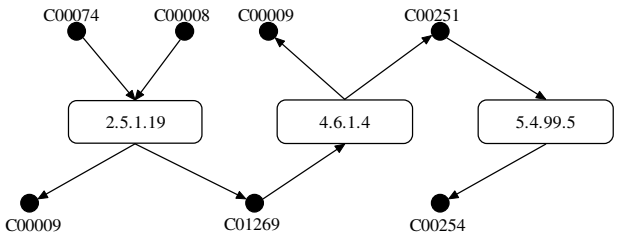


**Fig. 1.** Part of the aromatic pathway for Yeast

However, this representation does not incorporate any further information about the reactions. For example, there can be competing reactions if two enzymes elaborate the same input metabolites and the occurring of any of the reactions determines a certain sequence of successive reactions instead of another. Hence, it is important to know which reaction among the two is more probable to happen. Moreover, input metabolites not always are available. Their absence can cause a certain reaction not to occur and give rise to another sequence in the metabolic pathway. Therefore, it is crucial to know how probable a certain reaction is. This situation can be modeled by attaching to each reaction the probability that it happens. This requires a first-order representation framework that can handle for each predicate that expresses a reaction the probability that the predicate is true.

The simple incorporation of probabilities is not enough to model complex metabolic networks. The probabilities of the reactions depend on many factors, such as initial quantity of input metabolites, changes in the physical-chemical environment surrounding the cell and many more. For this reason it is a hard task to observe all the states of the biological machinery and try to assign probabilities to reactions. Therefore there is a need for machine learning methods that can learn distribution of probabilities from observations.

In order to model metabolic networks, two tasks must be performed. First, a relational model that describes the reactions must be built. This can be constructed manually by the expert or learned automatically by machine learning multi-relational methods. Once the model has been built, the second task is the assignment of probabilities. In this paper we do not deal with model building. We use the metabolic pathway built in [4] and model this pathway in a hybrid symbolic-statistical framework in order to automatically learn the probabilities of the reactions. A logic program that describes this metabolic pathway has been given in [5]. We extend this logic program in order to statistically model the pathway in the framework PRISM. After modeling the pathway in PRISM we perform learning of probabilities and show through experiments the feasibility of accurately learning reactions probabilities from metabolomics data using PRISM.

## 3 The Symbolic-Statistical Framework PRISM

PRISM (PRogramming In Statistical Modelling) [6] is a symbolic-statistical modeling language that integrates logic programming with learning algorithms for probabilistic programs. PRISM programs are not only just a probabilistic extension of logic

programs but are also able to learn from examples through the EM (Expectation-Maximization) algorithm which is built-in in the language. The parameter learning algorithm [7], provided by the language, is a new EM algorithm called graphical EM algorithm that when combined with the tabulated search has the same time complexity as existing EM algorithms, i.e. the Baum-Welch algorithm for HMMs (Hidden Markov Models), the Inside-Outside algorithm for PCFGs (Probabilistic Context-Free Grammars), and the one for singly connected BNs (Bayesian Networks) that have been developed independently in each research field. Since PRISM programs can be arbitrarily complex, all the formalisms such as HMMs, PCFGs and BNs can be described by these programs.

PRISM programs are defined as logic programs with a probability distribution given to facts that is called basic distribution. Formally a PRISM program is $P = F \cup R$ where $R$ is a set of logical rules working behind the observations and $F$ is a set of facts that models observations' uncertainty with a probability distribution. Through the built-in graphical EM algorithm the parameters (probabilities) of $F$ are learned and through the rules this learned probability distribution over the facts induces a probability distribution over the observations. As an example, we present a hidden markov model with two states slightly modified from that in [7]:

```
values(init,[s0,s1]).       % State  initialization
values(out(_),[a,b]).       % Symbol emission
values(tr(_),[s0,s1]).      % State transition
hmm(L):-                    % To observe a string L:
str_length(N),              % Get the string length as N
msw(init,S),        % Choose an initial state randomly
hmm(1,N,S,L).       % Start stochastic transition (loop)
hmm(T,N,_,[]):- T>N,!.      % Stop the loop
hmm(T,N,S,[Ob|Y]) :-        % Loop: state S,time  T
msw(out(S),Ob),             % Output Ob at the state S
msw(tr(S),Next),            % Transit from S to Next.
T1 is T+1,                  % Count up time
hmm(T1,N,Next,Y).           % Go next (recursion)
str_length(10).             % String length is 10
set_params :- set_sw(init, [0.9,0.1]), set_sw(tr(s0),
[0.2,0.8]), set_sw(tr(s1), [0.8,0.2]),
set_sw(out(s0),[0.5,0.5]), set_sw(out(s1),[0.6,0.4]).
```

The most appealing feature of PRISM is that it allows the users to use random switches to make probabilistic choices. A random switch has a name, a space of possible outcomes, and a probability distribution. In the program above, msw(init,S) probabilistically determines the initial state from which to start by tossing a coin. The predicate set_sw( init, [0.9,0.1]), states that the probability of starting from state s0 is 0.9 and from s1 is 0.1. The predicate *learn* in PRISM is used to learn from examples (a set of strings) the parameters (probabilities of init, out and tr) so that the ML (Maximum-Likelihood) is reached. For example, the learned parameters from a set of examples can be: switch init: s0 (0.6570), s1 (0.3429); switch out(s0): a (0.3257), b (0.6742); switch out(s1): a (0.7048), b (0.2951); switch tr(s0): s0 (0.2844), s1 (0.7155); switch tr(s1): s0 (0.5703), s1 (0.4296).␣After learning these ML parameters, we can calculate the probability of a certain observation using the predicate *prob*:

prob(hmm([a,a,a,a,a,b,b,b,b,b]) = 0.000117528. This way, we are able to define a probability distribution over the strings that we observe. Therefore from the basic distribution we have induced a probability distribution over the observations.

## 4 PRISM Modeling of Aromatic Amino Acid Pathway of Yeast

The logic foundation of PRISM facilitates the construction of a representation of the metabolic pathway described in the previous section. Predicates that describe reactions remain unchanged from a language representation point of view. What we need to statistically model the metabolic pathway is the extension with random switches of the logic program that describes the pathway. We define for every reaction a random switch with its relative space outcome. For example, in the following we describe the random switches for the reactions in Fig. 1.

values(switch_rea_2_5_1_19,[rea_2_5_1_19( yes, yes, yes, yes ),rea_2_5_1_19(yes, yes, no, no)]).
values(switch_ea_4_6_1_4,[rea_4_6_1_4(yes, yes, yes),rea_4_6_1_4(yes, no, no)]).
values(switch_rea_5_4_99_5,[rea_5_4_99_5( yes, yes ),rea_5_4_99_5(yes, no)]).

For each of the three reactions there is a random switch that can take one of the stated values at a certain time. For example, the value *rea_2_5_1_19(yes, yes, yes, yes)* means that at a certain moment the metabolites C00074 and C00008 are present and the reaction occurs producing C00009 and C00251. While the other value *rea_2_5_1_19(yes, yes, no, no)* means that the input metabolites are present but the reaction did not occur, thus the products C00009 and C00251 are not produced. Below we report the PRISM program for modeling the pathway in Figure 1. (The complete PRISM code for the whole metabolic pathway can be requested to the authors).

```
enzyme('2.5.1.19',rea_2_5_1_19,[C00074,C03175],[C00009,
C01269]).
enzyme('4.6.1.4',rea_4_6_1_4,[C01269],[C00009,C00251]).
enzyme('5.4.99.5',rea_5_4_99_5,[C00251],[C00254] ).
can_produce(Metabolites,Products) :-
can_produce(Metabolites,[],Products).
can_produce(Metabolites,Stalled,Products) :-
(possible_reaction(Metabolites,Stalled,Name,Inputs,Outp
uts,Rest) ->
reaction_call(Reaction,Inputs,Outputs,Call),
rand_sw(Call,Value),
((Value == rea_2_5_1_19(yes,yes,yes,yes);
Value == rea_4_6_1_4(yes,yes,yes);
Value == rea_5_4_99_5(yes,yes))  ->
can_produce(Rest,Stalled,Products )
;
can_produce(Metabolites,[Reaction|Stalled],Product));
Products = Metabolites).
```

```
rand_sw(ReactAndArgs,Value):-
ReactAndArgs=..[Pred|Args],
(Pred == rea_2_5_1_19 ->msw(switch_rea_2_5_1_19,Value);
(Pred == rea_4_6_1_4 ->msw(switch_rea_4_6_1_4,Value);
(Pred == rea_5_4_99_5 -> msw(switch_rea_5_4_99_5,Value)
;
true))). % do nothing
```

In the following, we trace the execution of the program. The top goal to prove that represents the observations in PRISM is *can_produce(Metabolites,_,Products)*. It will succeed if there is a pathway that leads from *Metabolites* to *Products*, in other words if there is a sequence of random choices (according to a probability distribution) that makes possible to prove the top goal. The predicate *possible_reaction* controls among the first three clauses of the program, if there is a possible reaction with *Metabolites* in input. Suppose that at a certain moment *Metabolites* = [C00074,C00008] and thus the reaction can happen. The variables *Inputs* and *Outputs* are bounded respectively to [C00074,C00008] and [C00009,C01269]. The predicate *reaction_call* constructs the body of the reaction that is the predicate *Call* which is in the form: *rea_2_5_1_19 ( _,_,_,_ )*. This means that the next predicate *rand_sw* will perform a random choice for the switch. This random choice which is made by the built-in predicate *msw(switch_rea_2_5_1_19,Value)* of PRISM, determines the next step of the execution, since *Value* can be either *rea_2_5_1_19(yes, yes, yes, yes)* or *rea_2_5_1_19(yes, yes, no, no).* In the first case it means the reaction has been probabilistically chosen to happen and the next step in the execution of the program which corresponds to the next reaction in the metabolic pathway is the call *can_produce(Rest, Stalled, Products)*. In the second case, the random choice *rea_2_5_1_19(yes, yes, no, no)* means that probabilistically the reaction did not occur and the sequence of the execution will be another, determined by the call *can_produce(Metabolites, [Reaction|Stalled],Products).*

In order to learn the probabilities of the reactions we need a set of observations of the form *can_produce(Metabolites,_,Products).* These observations that represent metabolomic data, are being intensively collected through available high throughput instruments and stored in metabolomics databases. In the next section, we show that from these observations, PRISM is able to accurately learn reaction probabilities.

## 5 Experiments

The scope of the experiments is to show empirically that on a medium-sized metabolic pathway the learning of the probability distributions from metabolomics data is feasible in PRISM. In order to assess the accuracy of learning the probabilities of the reactions we adopt the following method. A probability distribution $P_1$ ,..., $P_M$ is initially assigned to the clauses of the logic program so that each reaction has a probability attached. We call these M parameters the true parameters. Then we sample from this probability distribution $S$ samples (observations) by launching the top goal *can_produce(Metabolites,_,Products).* Once that we have these samples, we replace the probabilities by uniformly distributed ones. At this point the built-in predicate *learn* of PRISM is called in order to learn from the samples. PRISM learns M new parameters

$P_1^{'}$,..., $P_M^{'}$ , that represent the learned reaction probabilities from the observations. In order to assess the accuracy of the learned $P_i^{'}$ towards $P_i$ we use the RMSE (Root Mean Square Error) for each experiment with *S* samples.

$$\text{RMSE} = \sqrt{\left( \sum_{i=1}^{M} \frac{(P_i - P_i^{'})^2}{M} \right)}$$

We performed experiments on two types of networks. In the first there are not alternative branches in the metabolic pathway. It means that starting from any node in the network there are not multiple paths to reach another node in the network. While in the second network we add an alternative path. For each network, we have performed different experiments with a growing number *S* of samples in order to evaluate how the number of samples affects the accuracy and the learning time. For each *S* we have performed 10 experiments in order to assess the standard deviation of RMSE for different experiments with the same number of samples.

**Table 1.** Experiments on the 2 networks

| *S* – Number of samples | Mean of RMSE on 10 experiments | | Standard Deviation. of RMSE on 10 experiments | | Mean learning time on 10 experiments (seconds) | |
|---|---|---|---|---|---|---|
| | Network 1 | Network 2 | Network 1 | Network 2 | Network 1 | Network 2 |
| 100 | 0,14860 | 0,18080 | 0,00013 | 0,000021 | 0,031 | 0,078 |
| 200 | 0,13377 | 0,14723 | 0,00001 | 0,000041 | 0,078 | 0,094 |
| 400 | 0,09909 | 0,11796 | $1,5 * 10^{-7}$ | 0,000308 | 0,079 | 0,156 |
| 600 | 0,08263 | 0,10471 | 0,00001 | 0,000458 | 0,094 | 0,182 |
| 800 | 0,07766 | 0,08317 | $7,2 * 10^{-7}$ | $2,2 * 10^{-7}$ | 0,098 | 0,141 |
| 1000 | 0,07200 | 0,07708 | 0,00006 | $8,9 * 10^{-7}$ | 0,104 | 0,172 |
| 2000 | 0,06683 | 0,07027 | 0,000014 | 0,000686 | 0,118 | 0,194 |
| 4000 | 0,06442 | 0,06672 | 0,00001 | $2,9 * 10^{-7}$ | 0,140 | 0,204 |
| 6000 | 0,05667 | 0,05768 | 0,000018 | 0,000351 | 0,156 | 0,219 |
| 8000 | 0,05279 | 0,05306 | 0,000106 | $5,3 * 10^{-7}$ | 0,182 | 0,266 |
| 10000 | 0,05164 | 0,05231 | 0,000037 | 0,000481 | 0,203 | 0,281 |

As Table 1 shows, we get better results in terms of accuracy as *S* grows and the learning time is very low considering that the two networks are of medium size where Network 1 and Network 2 contain respectively 21 and 25 reactions. As RMSE decreases we note a slight increase of the learning time. Comparing the two networks, we can see that on the second network RMSE and the learning time are greater than on the first network. This is due to more nodes to explore during learning as the same node can be reached in different ways. However, the experiments show that given metabolomics data, learning accurately reaction probabilities in PRISM is feasible.

In a related work [5], SLPs (Stochastic Logic Programs) [8] were applied to the same problem. The advantage of our approach stands in the parameter learning phase. Parameter estimation in SLPs [9] requires the intractable computation of a normalizing constant. In [9] it is shown that the approach of simply enumerating refutations in the SLD-tree is tractable only for small problems because it requires the exploration of the entire SLD-tree of the top goal. Moreover, for parameter learning of SLPs there have not yet been developed tabulation techniques such as in PRISM

where tabulated search greatly increases efficiency [7]. However, structure learning for SLPs has been dealt with in [10] (in [9] the structure is supposed to be learned by another method and it only applies the parameter estimation algorithm to the given structure), while structure learning for PRISM programs has not been attempted.

## 6 Conclusion

We have applied the hybrid symbolic-statistical framework PRISM to a problem of modeling metabolic pathways and have shown through experiments the feasibility of learning reaction probabilities from metabolomics data for a medium-sized network. To the best of our knowledge this is the first application of the framework PRISM to a problem in Systems Biology. Very good probability estimation accuracy and learning times validate the hybrid approach to a problem where both relations and uncertainty must be handled.

As future work, we intend to investigate larger networks and the problem of model building from observations. We believe PRISM fast learning algorithm will help in exploring larger metabolic networks in reasonable times.

## References

1. Kitano, H.: Foundations of Systems Biology. MIT Press, Redmond, Washington (2001)
2. Kriete, A., Eils, R.: Computational Systems Biology. Elsevier - Academic Press, Amsterdam (2005)
3. Page, D., Craven, M.: Biological Applications of Multi-Relational Data Mining. Appears in SIGKDD Explorations, special issue on Multi-Relational Data Mining (2003)
4. Bryant, C.H., Muggleton, S.H., Oliver, S.G., Kell, D.B., Reiser, P., King, R.D.: Combining inductive logic programming, active learning and robotics to discover the function of genes. Electronic Transactions in Artificial Intelligence 5-B1(012), 1–36 (2001)
5. Angelopoulos, N., Muggleton, S.H.: Machine learning metabolic pathway descriptions using a probabilistic relational representation. Electronic Transactions in Artificial Intelligence 6 (2002)
6. Sato, T., Kameya, Y., PRISM,: PRISM: A symbolic-statistical modeling language. In: Proceedings of the 15th International Joint Conference on Artificial Intelligence, pp. 1330–1335 (1997)
7. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. Journal of Artificial Intelligence Research 15, 391–454 (2001)
8. Muggleton, S.H.: Stochastic logic programs. In: de Raedt, L. (ed.) Advances in Inductive Logic Programming, pp. 254–264. IOS Press, Amsterdam (1996)
9. Cussens, J.: Parameter estimation in stochastic logic programs. Machine Learning 44(3), 245–271 (2001)
10. Muggleton, S.H.: Learning structure and parameters of stochastic logic programs. In: Proceedings of the 10th International Conference on Inductive Logic Programming, Springer, Berlin (2002)