Mining Time-series Sequences of Reactions for Biologically Active Patterns in Metabolic Pathways

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

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

Abstract. Large quantities of metabolic profiling data are being gathered intensively in the rapidly growing field of Metabolomics. However, such data, in order to provide knowledge, must be machine-explored by robust methods that deal with complexity and uncertainty. Symbolic machine learning methods have the power to model structural and relational complexity while statistical machine learning ones provide principled approaches to uncertainty modeling. In this paper, we apply a hybrid symbolic-statistical framework to mine time-series sequences of reactions for biologically active paths in metabolic networks. We show through experiments that our approach provides a robust methodology for knowledge discovery in Systems Biology.

1 Introduction

Metabolomics [1] is a rapidly growing field. Analytical techniques and instruments such as Mass Spectrometry (MS) and Nuclear Magnetic Resonance (NMR) for gathering and analyzing voluminous metabolic data are being intensively refined. MS is now able to detect molecules at concentrations as low as 10^{-18} molar, and high-field NMR can efficiently differentiate between molecules that are highly similar in structure. The main problem in this research area is the study of the metabolome [2] which represents the collection of all the metabolites in a biological organism. This set of molecules consists of metabolic intermediates, hormones and other signalling molecules, and secondary metabolites. All these represent the chemical fingerprints that every specific cellular process leaves behind. Thus, in order to understand how cells work it is important to explore the metabolome in a principled and robust manner. However, the separate study of the metabolome would not give a deep comprehension of the organism, because biological systems' behavior is determined by complex interactions between their building components. Therefore, an integrated approach to studying biological systems is necessary. This has given rise to the Systems Biology [3] approach to modeling biological phenomena. In Systems Biology the main problem is to uncover and model how function and behavior of the biological machinery are implemented through complex interactions among its building blocks. Metabolomics data provide precious traces of the cell's circuits functioning, hence it is highly important for the Systems Biology approach to integrate metabolomics for a deeper understanding of biological systems [4].

Since biological circuits are hard to model and simulate, many efforts [5] 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 and the possibility to discover biologically active paths. 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. It is through the metabolic networks that every single living organism carries out all its activities. Thus, pathway analysis is crucial to understand cell's behavior and machine learning methods, that are not limited to only simulate biological networks, are essential to infer knowledge from exponentially growing observation data gathered by high-throughput instruments.

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. 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 stochastic 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 contribution of this paper is at the intersection of Systems Biology, Metabolomics and Machine Learning. We apply a hybrid symbolic-statistical framework to the problem of modeling metabolic pathways and mining active paths from time-series data. We show through experiments the feasibility of mining significant paths from metabolomics data in the form of traces of sequences of reactions.

The paper is organized as follows. Section 2 describes the problem of modeling metabolic pathways and the necessity for symbolic-statistical machine learning. Section 3 describes the hybrid framework PRISM. Section 4 describes modeling in PRISM of the *Bisphenol A Degradation* pathway of *Dechloromonas aromatica*. Section 5 presents experiments on mining stochastically generated sequences of reactions for biologically active paths. Section 6 concludes discussing related and future work.

2 Metabolic Pathways

Metabolic pathways can be represented as graphs where each node represents a chemical compound and a chemical reaction corresponds to a directed edge labeled by a protein that catalyzes the reaction. Thus, there is an edge from one compound (metabolite) to another compound (product) if there is an enzyme that transforms the metabolite into product. Figure 1. shows part of the pathway of *Bisphenol A Degradation* in *Dechloromonas aromatica* extracted from KEGG[1] database. We have chosen this pathway from the KEGG because, as we can see from Figure 1, starting from one point in the pathway there are multiple paths that can be explored. Therefore, the task of mining biologically active paths is harder because more paths should be explored in order to discover the active ones.

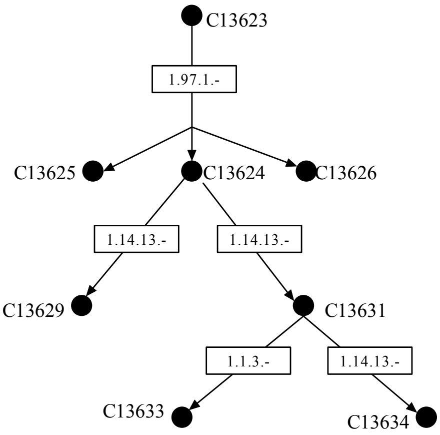

Fig. 1. Part of the metabolic pathway of *Bisphenol A Degradation* in *Dechloromonas aromatica*. The complete pathway is available from KEGG.

In order to model a metabolic pathway, a suitable framework for their simulation and mining must be able to handle relations. First-order logic representations have the

[1] http://www.genome.jp/kegg/

expressive power to model structural and relational problems. The metabolic pathway in Figure 1 can be easily represented in a first-order logic formalism as follows:

```
enzyme( 1.97.1.-, reaction_1_97_1_, [c13623], [c13625,c13624,c13626]).
enzyme( 1.14.13.-, reaction_1_14_13_a, [c13624], [c13629).
enzyme( 1.14.13.-, reaction_1_14_13_b, [c13624], [c13631]).
enzyme( 1.1.3.-, reaction_1_1_3, [c13631], [c13633]).
enzyme( 1.14.13.-, reaction_1_14_13_c, [c13631], [c13634]).
```

However, this representation does not incorporate any further information about the reactions. For example, as we can see there are two competing reactions because the enzyme *1.14.13.-* catalyzes two different reactions with the same chemical compound c13624 in input. Subsequently, two enzymes, *1.14.13.- and 1.1.3.-*, can elaborate the same input metabolites and thus two reactions compete among them. 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. The most probable reaction determines the biologically active path under certain conditions. This means that under certain conditions, a biological path becomes inactive or useless and another path may become active and yield different overall products in the whole pathway. The conditions under which the reactions happen, may change stochastically due to the random behavior of the biological environment. For example, some input metabolites can suddenly be not 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 conditions for the reactions to happen 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 under all the possible conditions and try to assign probabilities to reactions. Therefore there is a need for machine learning statistical methods that given certain conditions can learn distribution of probabilities from observations (the conditions here are meant as physical-chemical entities such as temperature, concentration of metabolites, entropy etc).

In order to model metabolic networks, two tasks must be performed. First, a relational model that describes the structure of the pathway must be build. There is already a large amount of accumulated knowledge about the structure of metabolic pathways such as that in KEGG and we can use all this background knowledge to skip the structure building process and concentrate on mining raw wet experimental-observational data. Indeed, graph structures are abundant but their main disadvantage in modeling cell's life is that they are static. This means that the pathway in Figure 1. does not express the stochastic dynamics in metabolic reactions. These graphs can be seen as useful static templates to interpret what can happen in the cell, but to faithfully reconstruct the cell's activity we must build a dynamic model that represents at a certain moment and under certain conditions what happens inside the cell. Thus, in order to mine biologically active patterns in the pathway under some conditions, we must first learn a dynamic-stochastic model from sequences of reactions that have been observed under those conditions. In order to confirm the feasibility of our approach of mining biological active patterns, we will proceed as follows. We will stochastically change the conditions for the reactions to happen (Section 5 describes how this is performed). Then, under each set of conditions, we stochastically generate sequences of reactions and finally after learning probability distributions for the reactions of the pathway, we perform mining for biological active patterns by querying the dynamic model we have built.

3 The Hybrid 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. PRISM represents a formal knowledge representation language for modeling scientific hypotheses about phenomena which are governed by rules and probabilities. 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 (no restriction on the form or size), the most popular probabilistic modeling 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 joint probability distribution over the set of least models of P, i.e. over the observations. This is called *distributional semantics* [8]. 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: current state is S, current time is 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 joint probability distribution over the observations.

4 Modeling *Bisphenol A Degradation* Pathway in PRISM

Since PRISM is a logic-based language, we can easily represent the metabolic pathway presented 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 Figure 1.

```
values(switch_rea_1_97_1, [rea_1_97_1( yes, yes, yes, yes), rea_1_97_1( yes, no, no,
no)]).
values(switch_rea_1_14_13_a,[rea_1_14_13_a(yes, yes), rea_1_14_13_a(yes, no)]).
values(switch_rea_1_14_13_b,[rea_1_14_13_b( yes, yes ),rea_1_14_13_b( yes, no)]).
values(switch_rea_1_1_3,[rea_1_1_3( yes, yes ),rea_1_1_3( yes, no)]).
values(switch_rea_1_14_13_b,[rea_1_14_13_c( yes, yes), rea_1_14_13_c( 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_1_97_1(yes, yes)* means that at a certain moment the metabolite c13623 is available and the reaction occurs producing the compounds c13623, c13624 and c13625. While the other value *rea_1_97_1(yes, no, no, no)* means that the input metabolite is present but the reaction stochastically did not occur , thus the products are not produced. Below we report the remaining part of the PRISM program for modeling the pathway in Figure 1. Together with the declarations in Section 2 for the possible reactions and those of the previous paragraph for the values of the random switches, the following logic program forms a model for stochastically modeling the pathway in Figure 1. (The complete PRISM code for the whole metabolic pathway can be requested to the authors).

```
produces( Metabolites, Products ) :-
      produces( Metabolites, [], Products ).

produces( Metabolites, Delayed, Products ) :-
      ( reaction( Metabolites, Name, Inputs, Outputs, Rest ) ->
               call_reaction( Reaction, Inputs, Outputs, Call ),
               rand_sw(Call,Value),
                        ((Value == rea_1_97_1( yes, yes, yes, yes );
                        Value == rea_1_14_13_a(yes, yes,);
                        Value == rea_1_14_13_b(yes, yes,);
                        Value == rea_1_14_13_c(yes, yes,);
                        Value == rea_1_1_3( yes, yes ))  ->
                        produces( Rest, Delayed, Products )
                        ;
                        produces(Metabolites, [Reaction|Delayed],Products)
               ;
               Products = Metabolites
      ).

rand_sw(ReactAndArgs,Value):-
      ReactAndArgs =..[Predicate|Arguments],
      (Predicate == rea_1_97_1-> msw(switch_rea_1_97_1,Value) ;
      (Predicate == rea_1_14_13_a ->  msw(switch_rea_1_14_13_a,Value);
      (Predicate == rea_1_14_13_b ->msw(switch_rea_1_14_13_b,Value);
      (Predicate == rea_1_14_13_c ->msw(switch_rea_1_14_13_c,Value);
      (Predicate == rea_1_1_3 ->msw(switch_rea_1_1_3,Value)
      ;
      true))))).  % do nothing
```

In the following, we trace the execution of the above logic program. The top goal to prove that represents the observations (sequences of reactions vastly produced by high-throughput technologies) for PRISM is *produces(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 *reaction* controls among the first clauses of the program, if there is a possible reaction with *Metabolites* in input. Suppose that at a certain moment *Metabolites* = [c13624] and thus two competing reactions can happen. Suppose one of the reaction is stochastically chosen and the variables *Inputs* and *Outputs* are bounded respectively to [c13624] and [c13629]. The predicate *call_reaction* constructs the body of the reaction that is the predicate *Call* which is in the form: *rea_1_14_13_a(_,_,_,_).* This means that the next predicate *rand_sw* will perform a random choice for the switch *switch_rea_1_14_13_a*. This random choice which is made by the built-in predicate *msw(switch_rea_1_14_13_a, Value)* of PRISM, determines the next step of the execution, since *Value* can be either *rea_1_14_13_a(yes, yes)* or *rea_1_14_13_a(yes, 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 *produces(Rest, Delayed, Products).* In the second case, the random choice *rea_1_14_13_a(yes, no)* means that probabilistically the reaction did not occur and the sequence of the execution will be another, determined by the call *produces(Metabolites, [Reaction|Delayed],Products)* which will try stochastically to choose the competing reaction catalyzed by the same enzyme *1.14.13.-* that given the same input c13624 produces the compound c13631. If this reaction occurs, then the next reaction in the sequence will be one of the competing reactions with c13631 as input.

In order to learn the probabilities of the reactions we need a set of observations of the form *produces(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 through the built-in graphical EM algorithm.

5 Mining Stochastically Generated Sequences of Reactions

A certain metabolic path becomes inactive or useless under certain conditions if a certain intermediate reaction in the path cannot occur under those conditions. In this paper we are not interested in the conditions themselves (these usually are stoichiometrics constraints). What is important for our purpose here, is that the conditions evolve stochastically. This means that by simulating various conditions we make possible a set of reactions instead of another, i.e. each set of conditions gives rise to a set of possible reactions that render some paths in the metabolic pathway biologically active and others biologically inactive under those conditions. In order to simulate various conditions, for each experiment we randomly assign probabilities to reactions. These probabilities represent the switches probabilities in PRISM. Thus, we have for each single experiment a set of conditions under the form of assigned reactions' probabilities (as probabilities are randomly generated and some of them may be equal to zero or in the range [0,9 - 0,999], among competing reactions one of them may not occur and this will cause some paths in the metabolic pathway to be inactive). The model constructed in this manner reflects the state of the biochemical environment under the given conditions at a certain moment. When the reactions happen, what is caught by a high-throughput instrument is a set of metabolites concentrations and their changes. For example, if a certain reaction happens then the concentration of the input metabolites decrease and that of the product compounds increase. This change is registered as a reaction, therefore catching all the time-series changes in concentration (this is actually performed intensively and accurately by current high-throughput technologies), means registering a time-series sequence of reactions. These constitute our mining data in order to re-construct biological active and inactive paths. By simulating the built model (this corresponds to simply running the PRISM program by calling the goal *produces(InputMetabolites*, *Products)* where *InputMetabolites* is a bounded list with the input compounds and *Products* is a logic variable that will be bounded to the list of product compounds yielded by the series of reactions), we will have time-series sequences of reactions as if we were observing the model by high-throughput instruments.

In order to evaluate the validity of our approach we have proceeded as follows. For each experiment (each experiment has a different set of conditions, i.e. probabilities of random switches that are stochastically assigned) we have stochastically generated sequences of reactions by sampling from the previously defined model. This is made possible by the predicate *sample* of PRISM. Once the sequences have been generated, we launch the predicate *learn* of PRISM to learn the probability of each random switch from the sequences. Once the model has been reconstructed we query it over the sequences and mine biologically active paths with the predicate *hindsight(Goal)* where *Goal* is bounded to the top-goal *[InputMetabolites,Products]*. With this predicate we get the probabilities of all the sub-goals for the top-goal *Goal*. If any of these probabilities is equal to zero then the relative path of the sub-goal is biologically inactive under the given conditions. The relative path can be extracted by the predicate *probf(SubGoal,ExplGraph)* where *ExplGraph* (explanation graph in PRISM) represents the explanation paths for *SubGoal*.

The accuracy of mining the sequences of reactions for biologically active patterns, depends on the ability to faithfully recontruct the model from the sequences. In order to assess the accuracy of learning the probabilities of the reactions and mining really biologically active paths we adopt the following method to evaluate the learning phase for the approach of the previous paragraph. We call the initial probability distribution P_1 ,.., P_M (that represents the conditions) assigned to the clauses of the logic program the true probability distribution and call the *M* parameters the true parameters. Once the sequences have been stochastically generated by this model, we forget the true parameters and replace their probabilities by uniformly distributed ones. When learning starts, PRISM learns *M* new parameters $P_1^{'}$,.., $P_M^{'}$, that represent the learned reaction probabilities from the sequences. In order to assess the accuracy of the learned $P_i^{'}$ towards P_i we use the RMSE (Root Mean Square Error) for each single experiment with *S* sequences.

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

In this way we can measure the difference between the actual observations and the response predicted by the model. We have performed different experiments with a growing number *S* of sequences in order to evaluate how the number of sequences affects the accuracy and the learning time. Moreover, we wanted to test also large datasets of sequences in order to provide a robust methodology since real metabolomics datasets are in general highly voluminous. For each *S* we have performed 100 experiments where for each experiment the set of conditions is stochastically generated as presented above. Table 1. reports for each *S* the RMSE and the learning time on average for 100 experiments. We have used the version 1.10 of the system PRISM on a Pentium 4, 2.4GHz machine.

Table 1. RMSE and learning time on average for 100 experiments

S – Number of sequences	Mean of RMSE on 100 experiments	Mean learning time on 100 experiments (seconds)
100	0,13932	0,047
200	0,13593	0,068
500	0,12999	0,090
1000	0,10405	0,125
2000	0,09685	0,297
4000	0,08676	0,484
8000	0,06808	0,547
15000	0,05426	0,612
30000	0,03297	0,695
50000	0,02924	0,735
100000	0,02250	1,172

As Table 1 shows, the learning accuracy increases as more data are available and due to the tabulation techniques in PRISM, learning times increases reasonably as data dimension grows significantly. The accuracy of learning can be evaluated as good for a number of sequences between 1000 and 15.000 and excellent for a number of sequences greater than 15.000 considering that the range where probabilities fall is [0,..,1] and the RMSE is under 0,05. This means that the paths have been faithfully reconstructed from the sequences and thus the predicates *hindsight* and *probf* in PRISM faithfully produce the biologically active paths in the pathway. Indeed, from empirical observations, we noted that all the queries performed by these two predicates reflected the real biological paths that are supposed to have produced the sequences. For instance, we noted that anytime the probability of the reaction catalyzed by the enzyme *1.14.13.-* (with input the compound c13624 and output c13631) was stochastically assigned to be too low (from 0 to 0.05) by the conditions generation phase, then the path that involves one of the two next reactions, the one catalyzed by the enzyme *1.1.3.-* and producing in output c13633, was mined as a biologically inactive path for the given conditions. Moreover, we noted for all the experiments that by slightly changing the conditions, many inactive paths became suddenly active and vice versa. This is quite interesting since it means that we can learn from sequences how conditions evolve in order to understand what changes them and what governs their randomness.

6. Discussion and Future Work

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 and mining biologically active paths from time-series sequences of reactions. The power of the proposed approach stands in the description language that allows to model relations and in the ability to model uncertainty in a robust manner. Moreover, we have also shown that the symbolic-statistical framework PRISM can be used as a stochastic simulator for biochemical reactions.

The most important related work is that in [9] where a probabilistic relational formalism is used for modeling metabolic networks. The PRISM program we have presented here is syntactically quite similar to the logic program in [9], but is semantically different in the way probability distributions are defined. Stochastic Logic Programs (SLPs) [10], used in [9], assign probabilities to clauses and define probability distributions on Prolog proof trees, while PRISM programs are based on the *distributional semantics* [8] and assign probabilities to atoms as we explained in Section 3. Most of other related work is not based on symbolic-statistical approaches. In [11, 12], graph-theory based approaches are used to find common or unique subgraphs in different pathway graphs to understand better why and how pathways differ or are similar. Other approaches are those that focus on text mining for metabolic pathways [13]. These methods have been applied to the voluminous literature on metabolic pathways to discover knowledge about the structure of the pathways. Text mining techniques focus on the structure building process trying to identify, in the accumulated experience about metabolic pathways, significant structural properties. Other approaches attempt to only stochastically simulate biochemical processes such StochSim [14] or FluxAnalyzer [15]. These are powerful tools to model the dynamic nature of cells for simulation purposes but lack machine learning abilities to infer knowledge from observations.

Although we have been able to reconstruct the model from the sequences of reactions, our approach is far from completing the real picture of a biochemical network. Much work remains to be done. First of all, we have not considered stoiochiometrics constraints which express quantitative relationships of the reactants and products in chemical reactions. We believe that adding these constraints to our approach will help reproduce better models. Another direction for future work regards plugging in the model other sources of data. Considering multiple sources of data can lead to better models in modeling metabolic pathways [16]. In PRISM this is straightforward because relational problems can be easily modeled due to the logic-based language. Another challenge is learning from incomplete raw metabolomic data. EM algorithms [17] are the state-of-the art for learning in the presence of missing data and since the graphical EM algorithm [7] that PRISM uses, is a version of this class of learning algorithms, we believe this will help in dealing with incomplete real datasets. In addition, in this paper we have considered a medium-sized metabolic pathway. For future work we intend to model very large metabolic pathways and hierarchical metabolic networks to see how the learning algorithms in PRISM scales for large datasets. We think the tabulation techniques used in PRISM will greatly help in dealing with a high number of paths to be explored. We also plan to investigate other important problems using the symbolic-statistical framework PRISM and other learning capabilities such as inductive relational learning for inferring missing pathways in existing metabolic networks or reconstructing whole novel pathways from sequences of observations.

References

1. Harrigan, G. G. and Goodacre, R. (eds). *Metabolic Profiling: Its Role in Biomarker Discovery and Gene Function Analysis*. Kluwer Academic Publishers, Boston (2003).
2. Oliver, S. G., Winson, M. K., Kell, D. B. and Baganz, F. Systematic functional analysis of the yeast genome. *Trends Biotechnol.* 16 (10): 373–378, (1998).
3. Kitano, H. (editor). *Foundations of Systems Biology.* MIT Press (2001).
4. Weckwerth, W. Metabolomics in systems biology. *Annu. Rev. Plant Biol*. 54, 669–689 (2003).
5. Kriete, A. Eils, R. *Computational Systems Biology.*, Elsevier - Academic Press (2005).
6. Sato, T. and Kameya, Y. PRISM: A symbolic-statistical modeling language. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence* , pp.1330–1335, (1997).
7. Sato, T. and Kameya, Y. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, Vol.15, pp.391–454, (2001).
8. Sato, T. A statistical learning method for logic programs with distribution semantics. In Leon Sterling, editor, *Proc. Twelfth International Conference on Logic Programming*, pages 715-729, Kanagawa, Japan, MIT Press, (1995).
9. Angelopoulos N. and Muggleton S.H. Machine learning metabolic pathway descriptions using a probabilistic relational representation. *Electronic Transactions in Artificial Intelligence*, 6, (2002).
10. Muggleton, S.H. Stochastic logic programs. In L. de Raedt, editor, *Advances in Inductive Logic Programming*, pages 254-264. IOS Press, (1996).
11. Koyuturk, M., Grama, A., and Szpankowski, W. An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks, Bioinformatics, Suppl. 1*: Proc. 12th Intl. Conf. Intelligent Systems for Molecular Biology* (ISMB'04) , 200-207, (2004).
12. You, C.H., Holder. L.B., and Cook: J. Application of Graph-based Data Mining to Metabolic Pathways. *Workshop on Data Mining in Bioinformatics*, ICDM, (2006).
13. Hoffmann, R., Krallinger, M., Andres, E, Tamames, J., Blaschke, C., Valencia, A. Text mining for metabolic pathways, signaling cascades, and protein networks *Sci STKE* 283, 21 (2005).
14. Le Novère, N. & Shimizu, T. S. StochSim: modelling of stochastic biomolecular processes. *Bioinformatics* 17, 575-576, (2001).
15. Klamt S, Stelling J, Ginkel M and Gilles ED. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics* 19(2): 261-269, (2003).
16. Fiehn, O. Combining genomics, metabolome analysis, and biochemical modelling to understand metabolic networks. *Comp. Funct. Genomics* **2** (3): 155–168, (2001).
17. Dempster, A. P., Laird, N. M., and Rubin, D. B.. Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society*, B39(1), 1{38, (1977).