Link and Node Prediction in Metabolic Networks with Probabilistic Logic

Angelika Kimmig[1] and Fabrizio Costa[2,⋆]

[1] Departement Computerwetenschappen, K.U. Leuven
Celestijnenlaan 200A - bus 2402, B-3001 Heverlee, Belgium
angelika.kimmig@cs.kuleuven.be
[2] Institut für Informatik, Albert-Ludwigs-Universität,
Georges-Koehler-Allee, Geb 106, D-79110 Freiburg, Germany
costa@informatik.uni-freiburg.de

Abstract. Information on metabolic processes for hundreds of organisms is available in public databases. However, this information is often incomplete or affected by uncertainty. Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore needed. To this end such systems should exploit data available from related organisms and cope with heterogeneous sources of information (e.g. phylogenetic relations). Here we start to investigate two fundamental problems concerning automatic metabolic networks curation, namely *link prediction* and *node prediction* using ProbLog, a simple yet powerful extension of the logic programming language Prolog with independent random variables.

1 Introduction

Living organisms rely on a large interconnected set of biochemical reactions to provide the requirements of mass and energy for the cellular processes to take place. This complex set of reactions constitute the organism's metabolic network [1]. Highly specialized proteins, called enzymes, are used to regulate the time and place for the various processes as most of the reactions taking place in organisms would be too slow without them. Enzymes control in practice which parts of the overall metabolic network is active in a given cell region in a given cellular phase. A large quantity of information about these networks accumulated through years of research, and is nowadays stored and organized in databases allowing researchers to develop network based approaches to study organisms metabolism. There exist collections of metabolic networks for several hundreds of organisms (e.g., the Kyoto Encyclopedia of Genes and Genomes (KEGG) [2] or the BioCyc database [3]) where relations between genes, enzymes, reactions and chemical compounds are available and organized in collections called "pathways". The knowledge that we have of these relations is however incomplete (most annotation efforts fail to assign functions to 40-60% of the protein sequences [4]) and is affected by uncertainty (wrong catalytic function assignment,

⋆ FC was a postdoctoral fellow at K.U. Leuven while this work was initiated.

M.R. Berthold (Ed.): Bisociative Knowledge Discovery, LNAI 7250, pp. 407–426, 2012.

incomplete annotation (e.g., only one function of a multi-domain protein) or non-specific assignment (e.g., to a protein family)). Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore in dear need. However, in order to overcome the limitations of homology searches, it is paramount to make use of information from heterogeneous sources and to therefore encode all the available data into complex relational data bases (i.e., BisoNets [5]). Finally, to leverage the different amount of coverage for different organisms (i.e., there is more information regarding humans than for other vertebrates), a case-based approach that uses information on related organisms should also be employed. All these requirements raise the problem of how to integrate heterogeneous and uncertain sources of information in a principled way.

Although systems for reconstructing pathways from relevant gene sets [6] and filling pathway-holes [7] are known in literature, they do not offer sufficient flexibility when new additional sources of information become available or, more importantly, in case one needs to change the set of queries involved in the solution of a specific task.

We study an approach that satisfies these flexibility requirements by representing metabolic networks in the probabilistic logical framework ProbLog [8], a simple yet powerful extension of the logic programming language Prolog with independent random variables in the form of *probabilistic facts*. This allows us to easily include background knowledge affected by uncertainty, and to obtain an answer to several key questions by performing probabilistic inference in a principled manner.

In this work, we start to investigate some fundamental problems concerning automatic metabolic networks curation, namely: 1) *link prediction*, i.e., estimation of the degree of belief in a link between a gene and an enzyme, and 2) *node prediction*, that is, whether the existence of a certain enzyme (and its link to an unknown gene) has to be hypothesized in order to maintain the contiguity of a pathway. For both tasks, the key components of our probabilistic model are (1) a preliminary estimate of the degree of belief for an association between a gene G and an enzyme E in an organism O, (2) background knowledge BK on organisms related to O obtained from the KEGG database, and (3) a linear model that predicts the probability of the gene-enzyme relation $G - E$ for the organism O given the dataset BK. The features employed in the linear model are complex queries and the associated values correspond to the probability of the query in BK including the preliminary estimate. The parameters of the model encode the relevance of the query for the specific pair gene-enzyme. The core idea is to leverage the flexibility of ProbLog to define meaningful queries at a conveniently abstract level. We finally compute the probability of the gene-enzyme relation $G - E$ based on the queries that are satisfied with high probability and that are predicted to be relevant for $G - E$.

The chapter is organized as follows: in Section 2 we introduce the probabilistic logic framework ProbLog; in Section 3 we describe how we model the knowledge associated with the metabolic reconstruction tasks and how we query this model

for prediction; finally in Section 4 we present some initial empirical results on a specific pathway in yeast.

2 The Probabilistic Logic Environment: ProbLog

Our work uses ProbLog to model data and queries. ProbLog is a probabilistic extension of the logic programming language Prolog. It thus combines the expressivity of a first order modeling language with the ability to reason under uncertainty. In contrast to propositional graphical models (such as Bayesian Networks), connections between random variables in ProbLog can be specified on the first order level, thus avoiding the need of explicitly grounding all information a priori. This results in a higher level of abstraction and more flexibility in the specification of queries. In this section, we briefly illustrate the basic ideas of ProbLog by means of an example; for more details, we refer to [8].

The following ProbLog program[1] models a tiny fraction of the type of network considered in this chapter:

$$0.8 :: \texttt{ortholog}(\texttt{g1},\texttt{g2}). \quad 0.7 :: \texttt{ortholog}(\texttt{g1},\texttt{g3}).$$
$$0.6 :: \texttt{function}(\texttt{g1},\texttt{e1}). \quad 0.9 :: \texttt{function}(\texttt{g2},\texttt{e1}). \quad 0.5 :: \texttt{function}(\texttt{g3},\texttt{e1}).$$

With probability 0.8, genes `g1` and `g2` are orthologs, with probability 0.6, the enzymatic function of `g1` is `e1`, and so forth. One can now add background knowledge to the program to define more complex relations. For instance,

$$\begin{aligned}
&\texttt{edge}(\texttt{X},\texttt{Y}) \ :- \ \texttt{ortholog}(\texttt{X},\texttt{Y}).\\
&\texttt{edge}(\texttt{X},\texttt{Y}) \ :- \ \texttt{function}(\texttt{X},\texttt{Y}).\\
&\texttt{connected}(\texttt{X},\texttt{Y}) \ :- \ \texttt{edge}(\texttt{X},\texttt{Y}).\\
&\texttt{connected}(\texttt{X},\texttt{Y}) \ :- \ \texttt{edge}(\texttt{X},\texttt{Z}),\texttt{connected}(\texttt{Z},\texttt{Y}).
\end{aligned}$$

defines a simple general path relation in terms of the edges present in the network, whereas

$$\texttt{connected\_via\_ortholog}(\texttt{X},\texttt{Y}) \ :- \ \texttt{ortholog}(\texttt{X},\texttt{Z}),\texttt{function}(\texttt{Z},\texttt{Y}).$$

defines a specific type of connection from a gene via an ortholog gene to an enzymatic function.

More formally, a *ProbLog program* T consists of a set of labeled facts $p_i :: f_i$ together with a set of definite clauses encoding *background knowledge* (BK).[2] Each ground instance of such a fact f_i is true with probability p_i, that is, corresponds to a random variable with probability p_i. All random variables are assumed

[1] We use standard Prolog notation, that is, arguments starting with lower case letters are constants, those starting with upper case letters are variables, and a definite clause $\texttt{h} :- \texttt{b}_1,\ldots,\texttt{b}_n$ is read as "if the $\texttt{b}_i$ are all true, `h` is true as well".

[2] Uncertain clauses can be modeled by adding a probabilistic fact to the clause body.

to be mutually independent. The program thus naturally defines a probability distribution

$$P^T(L) = \prod_{f_i \in L} p_i \prod_{f_i \in L^T \setminus L} (1 - p_i)$$

over logic programs $L \subseteq L^T = \{f_1, \cdots, f_n\}$. The *success probability* of query q is then defined as

$$P_s^T(q) = \sum_{L \subseteq L^T : L \cup BK \models q} P^T(L). \tag{1}$$

It thus corresponds to the probability that q is *provable* in a randomly sampled logic program.

Given the example program above, one could now ask for the probability of a connection between `g1` and `e1`, that is, for the success probability of query `connected(g1,e1)`. As enumerating all possible programs (subgraphs in the example) is infeasible in most cases, ProbLog instead calculates success probabilities using all proofs of a query. The query `connected(g1,e1)` has three proofs in our example: one direct connection, and two connections involving an additional gene each, with probabilities 0.6, $0.8 \cdot 0.9 = 0.72$ and $0.7 \cdot 0.5 = 0.35$, respectively. As there are several subgraphs that contain more than one of these connections, we cannot simply sum the probabilities of proofs. This problem is also known as the *disjoint-sum-problem* or the two-terminal network reliability problem, which is #P-complete [9]. When calculating success probabilities from proofs, one has to take care to address this problem and to remove the overlap between proofs. In the example, this could be done by explicitly stating that proofs only add information if none of the previous ones are true. That is, the second proof via `g2` only adds to the probability if the direct connection is not present, and its contribution therefore needs to be reduced to $0.8 \cdot 0.9 \cdot (1 - 0.6)$. Similarly, the third proof only adds information if neither the first nor the second are true, resulting in an overall probability of

$$\begin{aligned} P_s^T(\texttt{connected(g1,e1)}) &= 0.6 + 0.8 \cdot 0.9 \cdot (1 - 0.6) && (2)\\ &\quad + 0.7 \cdot 0.5 \cdot (1 - 0.6) \cdot (1 - 0.8) && (3)\\ &\quad + 0.7 \cdot 0.5 \cdot (1 - 0.6) \cdot 0.8 \cdot (1 - 0.9) && (4)\\ &= 0.9272 \end{aligned}$$

Here, (2) lists the contributions of the first and second proof as explained above, (3) and (4) that of the third proof, split into the two possible causes for the second proof being invalidated, that is, `ortholog(g1,g2)` being false, or `ortholog(g1,g2)` being true, but `function(g2,e1)` being false.

While this disjoining approach is sound for any order of the proofs, it does not scale very well. In practice, ProbLog therefore represents all proofs of the query as a propositional formula, and then uses advanced data structures to calculate the probability of this formula; we refer to [8] for the technical details.

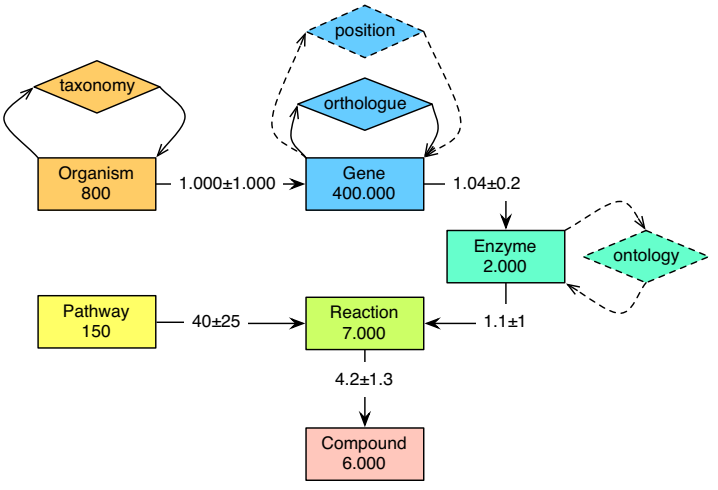

Fig. 1. Part of KEGG metabolic network used. The number in the node shape is the cardinality of the element set. The number on the edge is the average ± standard deviation number of relations between the element at the starting endpoint and the elements at the final endpoint of the edge. Dashed elements represent information present in KEGG but not currently used.

3 Method

We first discuss the information modeled in the background knowledge and then introduce the structural queries used in the prediction models.

3.1 Metabolic Network Representation

We represent the knowledge about metabolic networks in a probabilistic logical framework. To this end, we identify the main entities involved in the problem and encode all relations between them quantifying the uncertainty of each relation with an associated probability value. The entities that we consider (and that are represented as vertices in the global network) are: organisms, genes, enzymes, reactions, compounds (also called metabolites) and pathways (see Fig. 1).

Informally, a metabolic network contains information on the set of genes that belong to specific organisms and how these code for proteins, called enzymes, that are responsible for specific reactions involving the transformation of one compound into another. An organism is thus capable to perform certain related sets of reactions (semantically grouped under a single pathway concept) in order to produce and transform sets of metabolites, only if the organism can express the enzymes needed to catalyze those reactions.

We derive all data from the Kyoto Encyclopedia of Genes and Genomes (KEGG) [2].

Organisms are organized in a taxonomy with 5 levels and comprise eukaryotes (256) and prokaryotes (1332). As an example, in the KEGG taxonomy human would receive the following classification: Eukaryotes/ Animals/ Vertebrates/ Mammals/ Homo sapiens. We represent each level of the hierarchy as a node so to be able to express relationships between organisms that involve different degrees of relatedness. In this work we present results related only to the bacteria domain in prokaryotes.

The KEGG Release 58.0 (May 2011) lists 6,405,661 gene entries although in this work we limit the study to a subset of 400,000 genes relevant to the bacteria domain. Entry names of the KEGG GENES database are usually *locus-tags* given by the International Nucleotide Sequence Database Collaboration (INSDC) although a conversion into other gene/protein identifiers for main sequence databases such as NCBI and UniProt/Swiss-Prot, is possible. In this way additional information from external sources could be easily incorporated.

Enzymes are identified by the Enzyme Commission number (EC number) [10], which is a hierarchical classification scheme based on the chemical reactions they catalyze. Different enzymes in different organisms receive the same EC number if they catalyze the same reaction. Every enzyme code consists of the letters "EC" followed by four numbers separated by periods. Those numbers represent a progressively finer classification of the enzyme and induce a functional hierarchy. For example, the tripeptide aminopeptidases have the code "EC 3.4.11.4", whose components indicate the following groups of enzymes: EC 3 enzymes are hydrolases (enzymes that use water to break up some other molecule); EC 3.4 are hydrolases that act on peptide bonds; EC 3.4.11 are those hydrolases that cleave off the amino-terminal amino acid from a polypeptide; EC 3.4.11.4 are those that cleave off the amino-terminal end from a tripeptide.

The compounds involved in the metabolic transformations are a collection of small molecules, biopolymers, and other chemical substances that are relevant to biological systems. We consider 6000 unique compounds.

Enzyme mediated reactions between specific compounds are uniquely identified. The compounds involved in the reaction are distinguished into substrates and products. Note however that the reaction is considered to be bidirectional as we do not make use of more complex (and less reliable) reaction rate information.

Finally, the concept of *pathways* is used to express and organize our knowledge on metabolic processes occurring in a cell. A pathway is a set of related chemical reactions where a principal substance is modified by a series of chemical processes. Given the many compounds ("metabolites") and co-factors that are involved, single metabolic pathways can be quite complex. Moreover the separation in pathways is induced by human knowledge rather than being defined in a natural and uncontroversial way. Finally, the metabolic output of one pathway is the input for another, which implies that all the pathways are interconnected into the global complex metabolic network (see Fig. 2[3]).

All the aforementioned entities constitute vertices in our relational representation and are connected by several types of relations: at the highest level, the

[3] Image source: `http:\\commons.wikimedia.org\wiki\File:Metabolism_790px.svg`

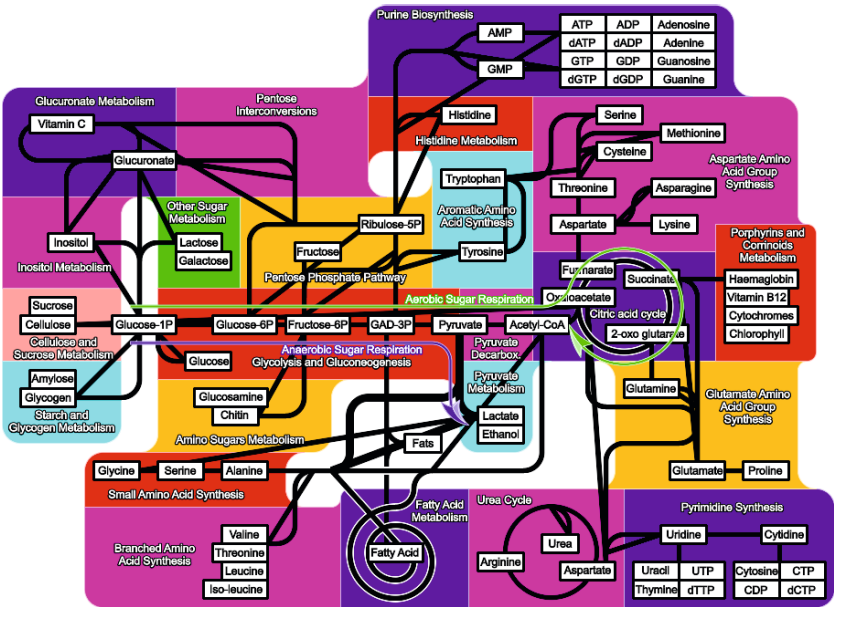

Fig. 2. Graphical representation of the major known pathways

various organisms are phylogenetically related to each other; genes are related to the organisms they are part of and they are related to each other via the ortholog relationship (see further in the text); enzymes are organized in a hierarchy following the Enzyme Commission number system; reactions are related to the compounds they require as substrate and to those they produce; genes are related to the enzymatic function of the protein that they code for; enzymes are related to the reactions they catalyze; and finally pathways are collections of related reactions. Our current model only treats the gene-enzyme relation probabilistically while all the other relations are assumed to be known with certainty. Note that in principle all relations are of the type many-to-many although in practice a gene is almost always associated to a single enzyme, which in turn catalyzes almost always a single reaction (see Fig. 1).

While the majority of these relations are intuitive, the ortholog relationship deserves some further detail. Orthologs, or orthologous genes, are genes in different species that are similar to each other because they descended from a single gene of the last common ancestor. Information about ortholog genes is available in KEGG and is obtained via a heuristic method that determines an ortholog cluster identifier in a bottom-up approach [11]. In this method, each gene subgroup is considered as a representative gene and the correspondence is computed using bi-directional best hit (BBH) relations obtained from the KEGG SSDB database which stores all-vs-all Smith-Waterman similarity scores. For efficiency reasons, the similarity score is thresholded and binarized: two genes are linked via

the ortholog relation only if each one is ranked in the top most similar genes of the other and if the similarity between the two exceeds a pre-specified threshold.

3.2 Models for Automatic Network Curation

Given the metabolic information about a set of organisms we identify two main problems of interest relevant for the concept of automatic network curation: 1) *link prediction*, where we estimate the probability associated to a given set of relations on the basis of an initial guess, in order to increase the consistency with respect to the information on related organisms; and 2) *node prediction*, where we introduce specific nodes in order to best fill gaps in the pathway of interest.

More in detail, we work in the following setting. We are given information about a new organism consisting of a set of genes and their associated functions (i.e., the enzyme they code for). This information is understood as being affected by uncertainty, and a probability serves as a preliminary approximation. Our goal is to derive more reliable estimates by integrating structural information from a broader context based on this first set of probabilities. The available background knowledge contains information on the metabolic network for a large set of organisms. In order to transfer knowledge from related organisms and/or genes we make use of two similarity notions: the first one is between the test organism and other organisms (obtained from the phylogenetic tree), the second between the genes in the test organism and genes in other organisms (via the ortholog relationship).

In principle we prefer evidence that is consistent across multiple sources as noise is likely to affect each source in an uncorrelated way. In practice, it is at times hard to propagate information from multiple sources because of the partial knowledge that we have of the metabolic network. In particular: a) not all genes of a test organism have an initial associated function; b) not all genes have known orthologs; c) not all reactions are known in a given pathway.

Another source of troubles in propagating evidence is to be found in the topological properties of the reaction network itself, known as the "small world" property [12]. A network is said to exhibit a small world property if there exist paths (reaction chains) of short length that can be followed to connect any two vertices (metabolites). This apparently surprising property of real metabolic networks can be explained by the presence of so called "currency" or "commodities" compounds [13], i.e., substances that occur commonly in any chemical process and that are assumed to be present in any needed quantity at any time in the cell environment. Common examples of such substances are water and ADP. Saying that two unrelated metabolites are connected because water is present in different reactions that involve them is therefore just an artifact of the data representation that has to be dealt with in an ad-hoc way. The problem is made non-trivial by the fact that there is no consensus on how to identify these substances. In this work we make use of the flexibility offered by the ProbLog language and specify a list of "accepted" (and "forbidden") compounds that can (cannot) be part of the path definition used to propagate information. Here we create such lists based on the frequency of the compounds in different reactions but expert knowledge can be as easily incorporated.

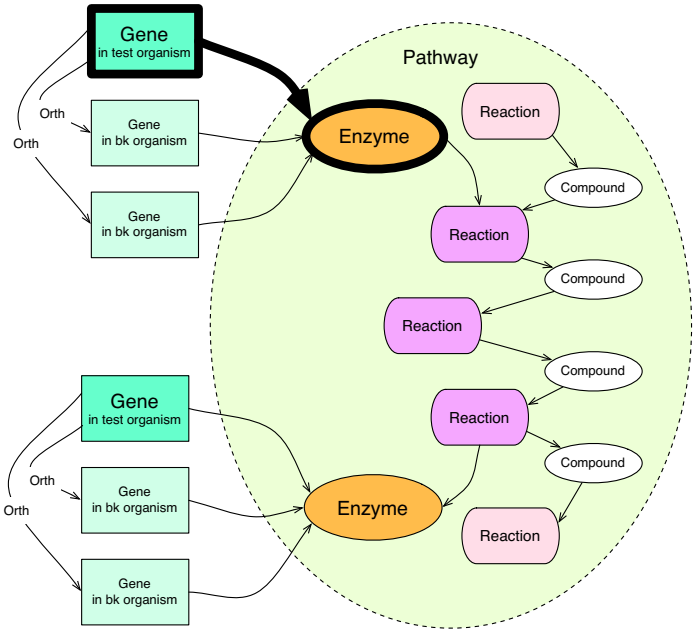

Fig. 3. Graphical representation of the portion of metabolic network used to obtain evidence for the link prediction task. The single gene-enzyme edge marked in bold corresponds to the substructures of type (1) used to obtain evidence for the link prediction task.

To summarize, the key idea of our prediction models is to use structural queries of increasing complexity to combine different forms of evidence. In the following, we discuss the queries we use for link prediction, their adaptation for node prediction, and the linear model that combines the success probabilities of the individual queries. Prediction then corresponds to a call to the ProbLog inference engine to compute the associated probability value.

Link Prediction Task. Figure 3 shows the part of the background knowledge queried to obtain support in link prediction. We use three types of queries of increasing complexity, illustrated in Figures 3, 4 and 5:

1. an estimate of the degree of belief for a gene-enzyme relation, either given a-priori or estimated by an external predictive system;
2. support coming from paths that contain the probabilistic gene-enzyme link under consideration; and
3. support coming from more complex subgraphs, that is, network portions that involve both the probabilistic gene-enzyme link and links to ortholog genes in related organisms.

For all queries, we only consider enzymes linked to a reaction in the pathway of interest. In particular, we require (2) to be a path that traverses in order the

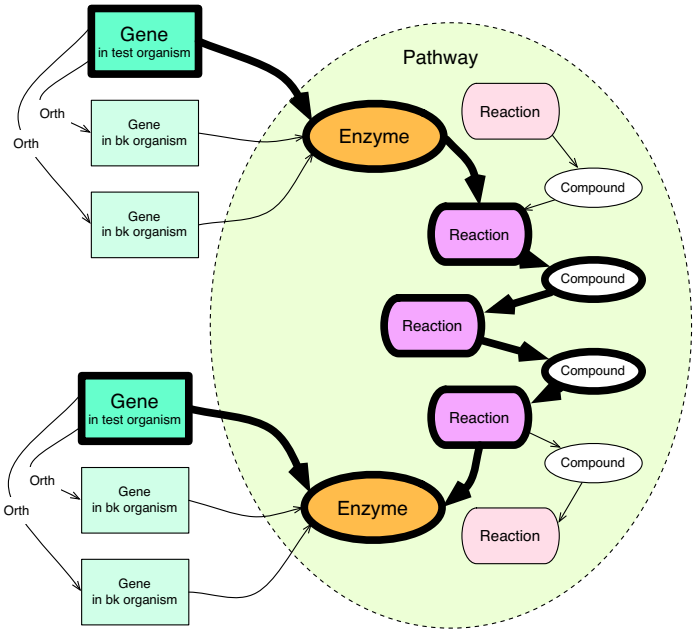

Fig. 4. Graphical representation of the substructures of type (2) used to obtain evidence for the link prediction task (marked in bold): path between two genes

following selected types of entities: gene, enzyme, reaction, compound, (reaction-compound)*, reaction, enzyme, gene. The intended meaning of the star notation here is that the path is only allowed to follow further reaction-compound links if the current reaction does not have an enzyme associated in the database. This latter condition is motivated by both computational efficiency issues (i.e., we do not consider all possible paths but only the shortest ones) and the desire to favor paths that make use of information relevant to the test organism. In words: we consider linear chains that originate in one gene of the test organism and end up in another gene of the same organism traversing the enzyme-reaction network relevant to a specific pathway. The subgraph for case (3) is obtained considering paths of type (2) with the addition of two extra paths at both ends. These provide additional links between the genes and enzymes at the end of the path via ortholog genes. The ratio here is to prefer evidence that is consistent with the information on similar genes in different organisms.

ProbLog allows us to specify the characteristics of these substructures at an intensional level. The network links are encoded using a set of (possibly probabilistic) predicates. Facts of the form `reaction_compound_reaction(r1,c,r2)` represent connections between reactions `r1` and `r2` via compound `c`. The list of compounds that may be traversed in queries is given as facts of the form `accept_compound(c)`. `ortholog(g1,g2)` facts list pairs of ortholog genes `g1` and `g2`, whereas `function(g,e)` facts link genes `g` to their enzymatic functions `e`.

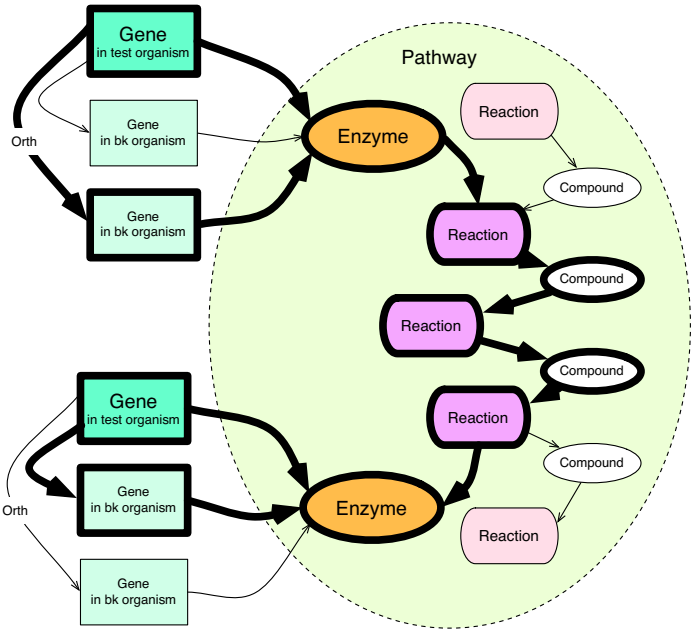

Fig. 5. Graphical representation of the substructures of type (3) used to obtain evidence for the link prediction task (marked in bold): subgraph involving ortholog genes

Finally, `reaction_enzyme(r,e)` facts connect reactions `r` in the background network to enzymes `e`. The background knowledge then defines additional relations and subgraph structures.

$$\texttt{reaction\_reaction}(\texttt{R1},\texttt{R2}) \ :- \ \texttt{reaction\_compound\_reaction}(\texttt{R1},\texttt{C},\texttt{R2}),\ \texttt{accept\_compound}(\texttt{C}).$$

restricts the reaction network to those links connected via accepted compounds as defined by the user.

$$\begin{aligned}\texttt{enzyme\_reaction\_path}(\texttt{G1},\texttt{E1},\texttt{E2},\texttt{G2}) \ :- \ & \texttt{function}(\texttt{G1},\texttt{E1}),\\ & \texttt{reaction\_enzyme}(\texttt{R1},\texttt{E1}),\\ & \texttt{reaction\_reaction}(\texttt{R1},\texttt{R2}),\\ & \texttt{reaction\_enzyme}(\texttt{R2},\texttt{E2}),\\ & \texttt{function}(\texttt{G2},\texttt{E2}).\end{aligned}$$

corresponds to the second query (modulo the star part), but making the gene and enzyme at the other end explicit, which is used in the third query to extend the query towards ortholog genes using

$$\texttt{ortholog\_support}(\texttt{G},\texttt{E}) \ :- \ \texttt{ortholog}(\texttt{G},\texttt{G2}),\texttt{function}(\texttt{G2},\texttt{E}).$$

The queries of Fig. 3-5 are then encoded as follows (where we omit some computational details for better readability):

```
query1(G,E) :- function(G,E), reaction_enzyme(R,E).
query2(G,E) :- enzyme_reaction_path(G,E,E2,G2).
query3(G,E) :- enzyme_reaction_path(G,E,E2,G2),
               ortholog_support(G,E), ortholog_support(G2,E2).
```

Note that if the database does not contain enough information to match a complex query, the query will simply fail. The failure does not provide any information and hence contributes a probability of 0. In these cases we resort to increasingly simpler queries in a fashion similar in spirit to the interpolation techniques employed in computational linguistics.[4]

Node Prediction Task. In node prediction, the goal is to identify enzymes that do not have an associated gene in the test organism, but would fill a hole in that organism's pathway if they did. As we cannot directly query the genes and enzymes of the organism of interest here, we resort to links between a hypothetical gene and the enzymes effectively present in the pathway of related organisms, cf. Fig. 6. We adapt the queries in Figures 3-5 as follows. Instead of the a-priori estimate of query type (1), which is not available here, we consider the average degree of belief in a link between the given enzyme and any known gene present in related organisms. For queries of types (2) and (3), we replace the test organism's gene at the top by a gene in some other related organism, but still require the path to end in a gene that is known to belong to the test organism.

Model. In both the link and node prediction setting, we estimate degrees of belief for our target relation by calculating the success probability (cf. Equation (1)) for each of the three types of supporting queries in the given model. We combine those results to answer the two main questions: 1) what is the probability of a specific gene of a test organism to be associated to a specific enzyme in the pathway? and 2) what is the probability of some unknown gene of a test organism to be associated to a specific enzyme in the pathway?

The combination is done via a linear model whose weights encode the reliability for each type of query.[5] Let $Q_i(G, E)$ be the success probability of the query of type i that relates the gene G with the enzyme E. The probability $p(G, E)$ that the gene effectively encodes the function E is computed as a convex combination of the success probability of each type of query, that is:

$$p(G, E) = \sum_{i=1,2,3} w_i(E) Q_i(G, E)$$

[4] When employing *n-gram* models, a common practice is to assess the probability of complex n-grams using the frequency counts of smaller n-grams that are more likely to occur in (small) datasets.

[5] Technically, the linear model is itself encoded as a ProbLog query and inference thus done in a single step without obtaining the individual success probabilities.

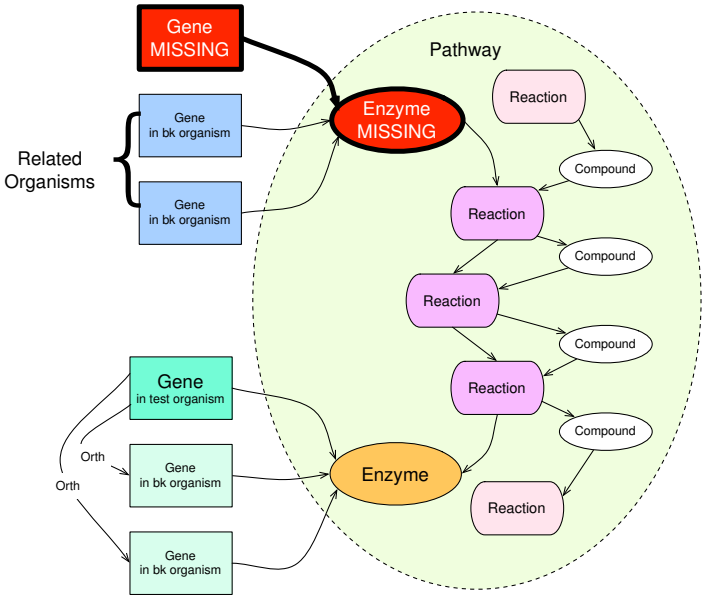

Fig. 6. Graphical representation of the portion of metabolic network used to obtain evidence for the node prediction task

where for each enzyme E, $\sum_{i=1,2,3} w_i(E) = 1$. We consider two variants of this model: one with enzyme-specific weights $w_i(E)$, and a global model that uses identical $w_i(E)$ for all enzymes.

The idea behind the linear model is to adapt to the level of missing information in the network: when assessing the degree of belief for an enzyme that is embedded in a network region where few reactions are known, it is better to trust the prior estimate with respect to more complex queries since they will mainly fail over the poorly connected reaction network; analogously when ortholog genes are known for a given enzyme, the evidence from the more complex queries becomes compelling. In summary, we adapt to the unknown local quality of the network by estimating the relative reliability of each query for the final answer on related organisms known in a background knowledge base.

In this work we explore two ways to induce the weights:

Frequency estimation: for each query type and enzyme, we count the number of proofs obtained for both positive and negative examples and obtain first estimates as $p/(p+n)$; these are then normalized over the three query types. Parameters for the global model, which does not model the dependency on enzymes, are obtained by summing counts over all enzymes before calculating frequencies.

Machine learning estimation: the weights are learned with ProbLog's gradient-descent approach to parameter learning [14]. Given a set of queries with associated target probabilities, this method uses standard gradient descent to minimize the *mean squared error* (MSE) on the training data.

4 Experimental Setup

Common sources of noise in available metabolic databases range from wrong catalytic function assignment to incomplete annotation (e.g., only one function of a multi-domain protein) or nonspecific assignment (e.g., to a protein family). In the empirical part of this study we analyze the curation/reconstruction capacity of the proposed system. To this end, we consider the KEGG data as ground truth and perturb the knowledge of the true function of a gene in such a way as to simulate these types of uncertainty in a controlled fashion.

4.1 Agnostic Noise Model

Since the enzymatic functions can be arranged in a hierarchical ontology [10], we can control the noise level by introducing extra links to enzymes that are in the neighborhood of the true enzymes. Two elements parametrize the noise model:

1. s: fraction of affected gene-enzyme pairs;
2. d: depth of lowest common parent in hierarchy.

We then proceed as follows: given an organism we select a fraction s of its known gene-enzyme links; for each link, we select all enzymes that have the lowest common parent with the link's enzyme at depth d in the hierarchy and that appear in the background knowledge network of the pathway of interest. We then introduce a uniform distribution over the set of gene-enzyme links resulting from the original gene and the selected enzymes.

General Setting. In the experiments reported here, we focus on the Pyruvate metabolism pathway (cf. Fig. 7) and organisms from subfamilies of proteobacteria, cf. Fig. 8. Pyruvate is an important intermediate in the fermentative metabolism of sugars by yeasts and is located at a major junction of assimilatory and dissimilatory reactions as well as at the branch-point between respiratory dissimilation of sugars and alcoholic fermentation.

A total of 40 organisms are picked uniformly at random, ensuring that all organisms of the smallest three subfamilies are included. For each such organism, we construct the background knowledge network by superimposing the networks of all organisms of the other five subfamilies, thus leaving out the most closely related organisms.

We create six different noise settings by perturbing the true relationships for $s = 1/5/10\%$ of gene-enzyme links, using $d = 2$ and $d = 3$, and use the linear model to rank candidate instances of the target relationship in each setting. For efficiency reasons, the linear model parameters are computed using the simple frequency estimate.

Experimental Results: Link Prediction. In the link prediction setting, positive examples are the test organism's real gene-enzyme links, while negative ones are the ones added by the noise model. The linear model uses the three queries

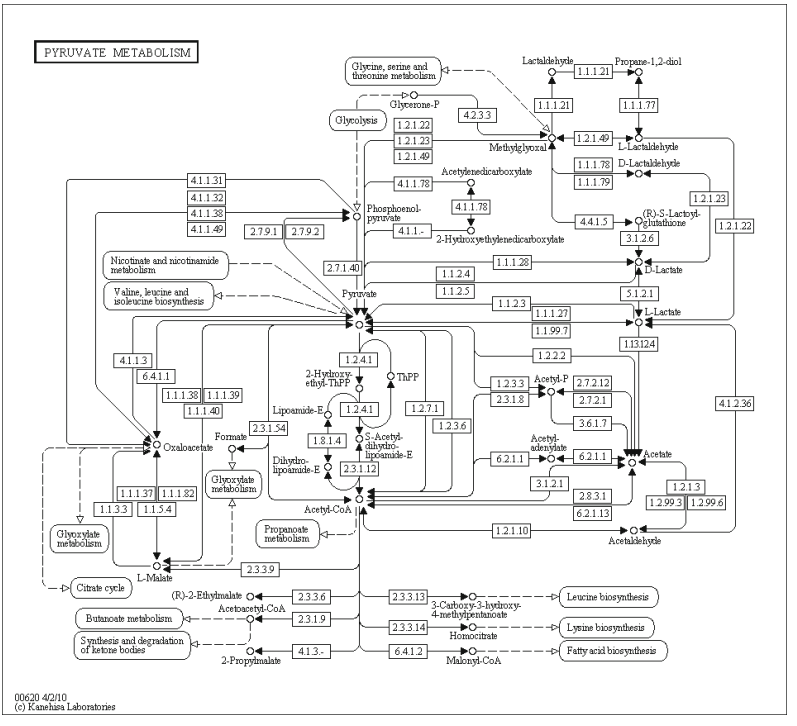

Fig. 7. Pyruvate metabolism pathway

depicted in Fig. 3-5. As the data is unbalanced, we report the area under the precision-recall curve as a performance measure. Results are summarized in Table 1 for the enzyme-specific linear model, the global mixture model, and the baseline using the most simple query type only. With increasing noise levels, the enzyme-based mixture model clearly improves over the baseline that does not take into account background information, and also over the less flexible global mixture model.

Experimental Results: Node Prediction. In the node prediction setting, examples are pairs of organisms and enzymes from the background knowledge. If the enzyme occurs in the organism's network, such an example is considered positive, and negative otherwise. We adopt an enzyme level leave-one-out design among those enzymes in the background knowledge that are not associated to any gene in the test organism. We remove these enzymes in turn and we measure the precision at one, that is, the fraction of times that the missing enzyme is ranked in first position as the most probable among all the missing enzymes.

The linear model uses the queries described in Section 3. Results are summarized in Table 2. While both mixture models significantly improve over the random ranking of all background enzymes, there is no significant difference between the global model (which doesn't take into account enzyme-specific in-

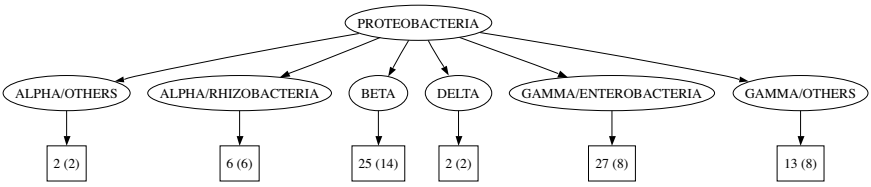

Fig. 8. Overview of organisms and subfamilies used in the background knowledge, including total number of organisms and number of organisms used as test cases (in brackets)

Table 1. Link prediction with varying noise level s and d: average and standard deviation of area under the precision-recall curve over 40 test organisms for the enzyme-specific linear model, the global mixture model, and the baseline using the most simple query type only

d=3			
s	enzyme	global	baseline
1%	0.987 ± 0.019	0.980 ± 0.026	0.975 ± 0.025
5%	0.935 ± 0.039	0.921 ± 0.045	0.911 ± 0.049
10%	0.863 ± 0.065	0.828 ± 0.068	0.831 ± 0.062
d=2			
s	enzyme	global	baseline
1%	0.981 ± 0.022	0.973 ± 0.027	0.966 ± 0.027
5%	0.889 ± 0.040	0.867 ± 0.045	0.853 ± 0.047
10%	0.775 ± 0.059	0.721 ± 0.064	0.743 ± 0.052

Table 2. Node prediction with varying noise level s and d: average and standard deviation of precision at one over 40 test organisms for the enzyme-specific linear model, the global mixture model, and the baseline using a random ranking

d=3			
s	enzyme	global	baseline
1%	0.218 ± 0.111	0.271 ± 0.143	0.020 ± 0.000
5%	0.217 ± 0.091	0.340 ± 0.124	0.020 ± 0.000
10%	0.198 ± 0.082	0.325 ± 0.144	0.020 ± 0.000
d=2			
s	enzyme	global	baseline
1%	0.223 ± 0.107	0.290 ± 0.165	0.020 ± 0.000
5%	0.224 ± 0.045	0.386 ± 0.081	0.019 ± 0.005
10%	0.152 ± 0.035	0.262 ± 0.080	0.011 ± 0.010

formation) and the enzyme-specific model. We conjecture that averaging the performance over "easy" and "hard" to predict enzymes yields a too coarse result and that a more detailed analysis is needed to identify the conditions that favour the enzyme-specific vs. the global model.

4.2 Noise Model for Unreliable Predictions

In this scenario, we assume that a predictor (i.e., a machine learning algorithm) is available and that it can compute the enzymatic function of a gene with a certain reliability. Instead of working with a specific predictor here we perturb the knowledge of the true function of a gene in order to simulate different degrees of reliability. Once again we make use of the fact that the enzymatic functions can be arranged in a hierarchical ontology [10]. Under this assumption we relate the topological distance in the ontology tree to the functional distance, i.e., the closer two enzyme nodes are in the hierarchy the more similar their functions. Under this assumption we build a noise model described by the following parameters:

1. s: fraction of affected genes;
2. k: number of noisy gene-enzyme links added per gene;
3. σ_{EC}: parameter controlling the size of the neighborhood where to randomly sample the additional noisy gene-enzyme links;
4. σ_N: parameter controlling the quantity of noise added to the gene-enzyme relationship probability estimate.

We then proceed as follows (see Fig. 9). Given an organism, we select a fraction s of its genes. For each selected gene, we add k extra links to randomly sampled *nearby* enzymes. Sampling selects enzymes using a normal distribution $N(0, \sigma_{EC})$ over their topological distance induced by the ontology, i.e., the length of the shortest path between the leafs containing the actual and the sampled enzyme in the tree structured ontology. Finally, we obtain the degree of belief for the link between the gene and the randomly selected enzyme as the probability of selecting the enzyme plus additional $N(0, \sigma_N)$ noise. In this way enzymes that are less related to (i.e., more distant from) the true enzymatic function of the original gene receive on average a smaller probability.

Experimental Results. In the experiments reported here, we focus on the Pyruvate metabolism pathway for the Escherichia coli UTI89 test organism. We perturb the true relationships with k=5 extra links for $s = 50\%$ of genes. The probability estimate of the gene-enzyme relationship receives additional noise from $N(0, \frac{1}{8})$.

The linear model parameters are computed using ProbLog's gradient-descent approach to parameter learning [14]. We use default settings in our experiments and run learning for at most 50 iterations, stopping earlier if the MSE on the training data does not change between two successive iterations. Training data is generated from the other organisms with the same parent in the organism hierarchy as the test organism, and target probabilities are set to 1.0 for positive and 0.0 for negative examples, respectively.

In the link prediction setting, positive examples are real gene-enzyme links, while negative ones are the ones added by the noise model where no real one is known between these entities. We use the three queries depicted in Fig. 3-5. We measure the area under the precision-recall curve.

When using the initial (perturbed) estimate for the gene-enzyme link we achieve an AUCPR of 0.69. If we use only the most complex query (type (3))

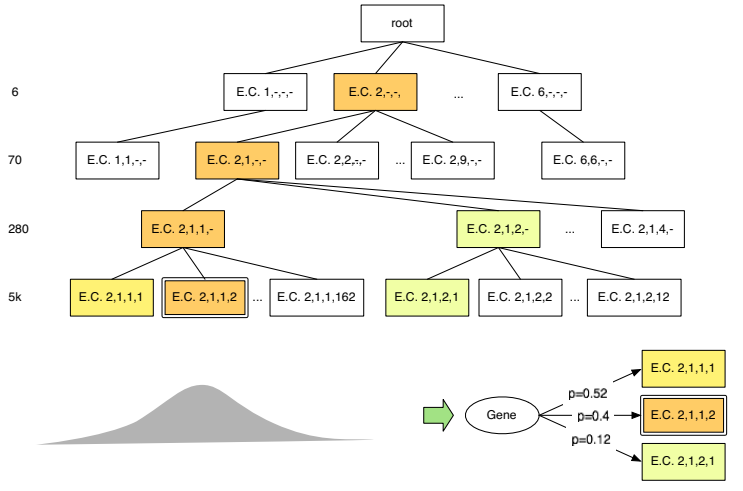

Fig. 9. Noise model: the E.C. hierarchy induced metric notion (i.e., topological distance between nodes) is used for the perturbed enzymatic function. The hypothetical true enzyme is marked with a double line. In the example a gene is associated to an incorrect enzymatic activity with probability 0.52 and to the correct one with probability 0.4.

we increase to 0.74, but when we learn the linear model over all queries we achieve 0.80. Note that simply learning a fixed mixture of experts for the whole organism (i.e., not modeling the dependency on the enzyme) we do not improve over the initial 0.69 result, as for this particular test organism, it is better to resort on average to the most simple query.

In the node prediction experiment, we follow the same scheme as above. That is, we adopt an enzyme level leave-one-out design among those enzymes in the background knowledge that are not associated to any gene in the test organism, remove these enzymes in turn and measure the precision at one.

The set of training examples is the set of all pairs of training organisms (as before) and enzymes appearing in the pathway for organisms different from the test organism. Such a pair is considered positive if the enzyme appears in the organism's pathway, and negative else.

We use the query described in Section 3 both with and without ortholog information, as well as a basic query that predicts each enzyme with the average probability of a gene-enzyme link involving this enzyme in one of the training organisms. In this experiment we achieve a precision at one of 0.66 over 35 possible enzymes (i.e., the baseline random guessing precision at one would be 0.03).

5 Conclusions

We have started tackling the problem of automatic network curation by employing the ProbLog probabilistic logic framework. To overcome the limitations of homology searches, we have made use of information from heterogeneous sources,

encoding all available data into a large BisoNet. To leverage the different quantity and quality of information available for different organisms, we have used a case-based approach linking information on related organisms. The use of a probabilistic logic framework has allowed us to: a) represent the knowledge about the metabolic network even when affected by uncertainty, and b) express complex queries to extract support for the presence of missing links or missing nodes in an abstract and flexible way. Initial experimental evidence shows that we can partially recover missing information and correct inconsistent information. Future work includes the integration of gene function predictor and the development of novel queries that make use of additional sources of information such as the gene position in the genome or the co-expression of genes in the same pathway from medical literature abstract analysis.

Acknowledgments. A. Kimmig is supported by the Research Foundation Flanders (FWO Vlaanderen). F. Costa was supported by the GOA project 2008/08 Probabilistic Logic Learning and by the European Commission under the 7th Framework Programme, contract no. BISON-211898.

References

1. Alm, E., Arkin, A.: Biological networks. Current Opinion in Structural Biology (13), 193–202 (January 2003)
2. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M.: KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research 38(Database issue), D355–D360 (2010)
3. Karp, P., Ouzounis, C., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V., Lopez-Bigas, N.: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research 19, 6083–6089 (2005)
4. Pouliot, Y., Karp, P.: A survey of orphan enzyme activities. BMC Bioinformatics 8(1), 244 (2007)
5. Kötter, T., Berthold, M.R.: From Information Networks to Bisociative Information Networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 33–50. Springer, Heidelberg (2012)
6. Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A., Kanehisa, M.: KAAS: an automatic genome annotation and pathway reconstruction server. Nucleic Acids Research 35(Web Server issue), W182–W185 (2007)
7. Green, M., Karp, P.: A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC Bioinformatics 5(1), 76 (2004)
8. Kimmig, A., Demoen, B., De Raedt, L., Santos Costa, V., Rocha, R.: On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming (TPLP) 11, 235–262 (2011)

9. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM Journal on Computing 8(3), 410–421 (1979)
10. Webb, E.C.: Enzyme nomenclature 1992: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes. Published for the International Union of Biochemistry and Molecular Biology by Academic Press, San Diego (1992)
11. Moriya, Y., Katayama, T., Nakaya, A., Itoh, M., Yoshizawa, A., Okuda, S., Kanehisa, M.: Automatic generation of KEGG OC (Ortholog Cluster) and its assignment to draft genomes. In: International Conference on Genome Informatics (2004)
12. Wagner, A., Fell, D.A.: The small world inside large metabolic networks. Proceedings of the Royal Society B: Biological Sciences 268(1478), 1803–1810 (2001)
13. Huss, M., Holme, P.: Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks. IET Systems Biology 1, 280–285 (2007)
14. Gutmann, B., Kimmig, A., Kersting, K., De Raedt, L.: Parameter Learning in Probabilistic Databases: A Least Squares Approach. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 473–488. Springer, Heidelberg (2008)