A Toolkit Supporting Formal Reasoning about Causality in Metabolic Networks

Chiara Bodei*[1], Andrea Bracciali[1] and Davide Chiarugi[2]

[1]Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3, I-56127, Pisa, Italy
[2]Dipartimento di Scienze Matematiche e Informatiche, Università di Siena, Pian dei Mantellini, 4, Italy

Email: Chiara Bodei*- chiara@di.unipi.it; Andrea Bracciali - braccia@di.unipi.it; Davide Chiarugi - chiarugi3@unisi.it;

*Corresponding author

Abstract

Background: Metabolic networks present a complex interconnected structure, whose understanding is in general a not trivial task. Several formal approaches have been developed to support the investigation of such networks, like quantitative models based on ODEs and machine learning techniques. One of the relevant problems in this context is the comprehension of *causality dependencies* amongst the molecules involved in the metabolic process.

Results: We propose a formal analysis approach aiming at featuring both expressiveness and ease of use. Its main ingredients are: *i)* a minimal notation to precisely represent bio-chemical interactions, and *ii)* an automated tool allowing the human expert to easily vary conditions of the *in silico* experiment. In particular, we exploit an analogy between logical implication and chemical reaction, i.e., roughly, the reaction of two molecules A and B producing a third one, C, can be interpreted as A and B logically imply C. Starting from a description of a metabolic network, in terms of reaction rules and initial conditions, chains of reactions, causally depending one from the another, can be mechanically deduced. Then, both the components of the initial state and, noticeably, the clauses ruling reactions can be changed and a new trial of the experiment started, according to a *what-if* investigation strategy. The method is supported by a computational logic counterpart, based on a Prolog implementation, which allows for a representation language closely correspondent to the adopted chemical abstract notation. The proposed framework has been validated by studying the robustness of

the metabolic network of *Escherichia coli* K12. Selected genes have been knocked-out by disabling the rules regarding the encoded enzymes. Results are coherent with the actual biological behaviour.

Conclusions: Starting from the presented work, our goal is to provide an effective analysis tool, supported by an efficient full-fledged computational counterpart, which can fruitfully drive *in vitro* experiments by effectively pruning non promising directions. More large-scale experiments are ongoing.

Background

In systems biology, the biological knowledge drives the development of models and the in silico analysis of these models supports the design of in vivo experiments, in a virtuous circle between empirical and theoretical investigation.

Actually, the models of complex systems code a lot of information and it is not easy to extract correlations or causal dependencies amongst the elements involved in the biological interaction networks. By rephrasing [1], "diagrams of interconnections represent a sort of static roadmaps, but what we really seek to know are the traffic patterns, why such patterns emerge, and how we can control them". Having a formal description of the interconnections and a methodology to perform software simulation on how these patterns are, should help in orientating wet-lab experimentation.

In this paper, we focus on "metabolic networks", i.e. on the set of all the cellular biochemical pathways involved in energy management and in the synthesis of structural components. Biochemical pathways are typically composed by chains of enzymatically catalyzed chemical reactions and are interconnected in a complex way. Thus the study of the overall behaviour of metabolic networks appears difficult with traditional experimental techniques, which often seem to offer inadequate tools to investigate such global properties. Indeed, whenever we think to a metabolic network, we just put together the components of the system under analysis: the role of each component is clear, whereas the overall behaviour of the whole system is not. Nevertheless, a deep understanding of the causal relations underlying the functioning of cellular metabolism appears to be a crucial task for biologists both for theoretical reasons and for the more applicative purposes of metabolic engineering. Under this regard, causality can play an important role, by finding chains of reactions that connect the parts of the system of

interest, e.g. determining correlations among molecules that are not apparently correlated.

We apply techniques from formal methods and from computational logic to develop a very abstract qualitative model of metabolic networks, where the focus is on causality. To this aim, we exploit an analogy between logical implications and chemical reactions, by interpreting the reaction of two molecules A and B producing a third one, C, as A and B logically imply C. We obtain a description of a metabolic network, in terms of reaction rules and initial conditions, from which we start to mechanically deduce chains of reactions, logically/causally depending one from the another. The framework appears profitable for biologists, because their usual representation language has a direct interpretation in our formalism, which, in turn, can be straightforwardly translated into an input for a suitable tool, that we have developed using standard logic programming techniques. What is more, this tool gives the opportunity to think about the model itself, by making it easy to vary both the components of the initial state and, noticeably, the clauses that rule reactions.

Our approach is a sort of "what-if" analysis, repeatedly exploring different scenarios, each one derived from a different set of hypotheses. Our tool allows us to rapidly evaluate the impact of changes in the hypotheses on a particular observable outcome. Thus, we obtain an interactive and effective analysis, that can be used to suggest which are the deductions that deserve to be tested in vitro, by pruning those that seem not to be promising.

Related Works and Comparison. A recent research line exploits well established theories and techniques of Formal Methods for Systems Biology, by using them to support the interpretation of the big amount of raw data now available for analysis.

In [2, 3], the authors apply a causal semantics of the π-calculus [4] to describe biochemical processes. The process computations that can be obtained quite accurately capture and reflect the real behaviour of biological systems and causality has a key role in enhancing precision in such simulations. Our starting point is quite similar, but our model is even more skeletal and abstract, based on descriptions of biological systems given in terms of molecular entities and reaction rules that specify their interactions and implicitly code the causes of reactions. Differently from precise behaviour descriptions, like the one based on process algebras such as π-calculus, logical deductions allows us to summarise possible pathway evolutions, of which causal chains form the backbone.

Another formal approach close to ours is that of Pathway logic (see e.g. [5, 6]), based on rewriting logic it can be fruitfully used to model biological processes. Rewrite rules describe local changes and the molecular patterns that cause them. Rules can be concurrently applied and this corresponds to the actual possibility of biological

compartments to independently evolve. This offers a basis for in silico experiments and for advanced forms of symbolic analysis. At present, in our framework, the concurrent aspect is deliberately ignored.

Other phenomena of metabolic networks may benefit from a logic-based representation, as done, for instance, in [7], with motivations similar to ours. That proposal is based on a combination of Abduction and Induction: abduction allows inference from observable effects (see also [8]) and therefore it is used to generate hypotheses, while induction has the aim of learning general rules from these abduced hypotheses. The predictive accuracy increases with the number of training examples. This methodology has a richer representation language than ours and aiming to address a different class of problems in a different experimental setting.

Also graph theory is exploited to model metabolic network. For instance, in [9], the authors focus on the topology of metabolic networks, by abstracting away from stoichiometric aspects. Networks are represented as graphs of metabolites and reactions. The idea underlying their work is not very far from ours, in terms of interests in chains of reactions. Nevertheless, their approach is clearly dynamic and graph theory offers different methodological features.

Results and Discussion

Representation language and logical interpretation

Several formal languages have been proposed to model aspects of biological interaction, like interaction sites or membrane compartments, e.g. [10–12]. Although they precisely capture the features of interest, it has seemed to us worth starting from the definition of a more abstract and intuitive "common" language, which, from the one hand is close to biochemical intuition, and from the other hand possess a straightforward computational counterpart. The underlying idea is to offer both to biologists and to computer scientists a simple and skeletal "lingua franca" to abstractly specify the desired network of causal dependencies. In particular, starting from a given chemical reaction in the form:

$$aA + bB \rightarrow cC + dD$$

we define a relation having the form:

$$A \circ B \rightarrow C \circ D$$

that is an abstraction of the chemical counterpart. This abstraction actually represents the possibility for C and D to be produced or caused by the presence of both A and B

(whose availability could in turn generate other species). Since we focus on causality relationships, we abstract away from features related to kinetics, thermodynamics and stoichiometry. Moreover we do not take into accout quantities, since we consider the needed amount of reactants always available. Finally, we also abstract from the actual dynamics of the studied system. Therefore, our representation is alternative to the standard notation of chemical reactions and it is tailored to catch those features we are interested in. Furthermore, it naturally leads towards an executable language.
More in general, we can define systems composed by an initial set I of elements that may cause new species according a set of rules, reflecting the known chemical reactions, as showed below.

$$I ::= A_1 \circ \ldots \circ \ A_n \qquad A \circ B \rightarrow C \circ \ldots \circ D \qquad A \rightarrow C \circ \ldots \circ D$$

Here $\circ$ stands for spatial co-location, and $\rightarrow$ for possible causal relation.

For instance, let us consider an abstraction of some reactions pertaining to glycolysis. In particular, let us model the phosphorilation of the Glucose done by the PTS-system (a bacterial membrane carrier) that lead to the production of Glucose 6-phosphate, and Pyruvate starting from Phosphoenolpyruvate and Glucose. With obvious mnemonic codes, this can be written as

$$glc \circ pep \ \rightarrow \ pyr \circ glc6p$$

$$pep \ \rightarrow \ pyr$$

Note that *pep* is involved into two rules, according to its multiple roles played in glycolysis. The second rule actually corresponds to the reaction catalysed by the pyruvate carboxylase enzyme, that is this rule corresponds to the activity caused by the gene coding for the enzyme; finally "removing" the rule has the same conceptual meaning of "knocking-out" the gene in the simulation.

From the computer science point of view, we need to address the task of providing a computational, i.e. executable, interpretation of our representation language. Each rule, $A \circ B \rightarrow C$ say, can be read as the logical implication $A \wedge B \Rightarrow C$. Moreover, the elements present in the initial state of the experiments can be understood as logical facts, i.e. premise-less clauses like $true \Rightarrow A$. This interpretation, for a quite large class of rules, viz. the Horn Clauses, has a well studied computational counterpart: Logic Programming [13]. The main difference of this format rule with our notation is that implications can have only one element in the head, so that the above rule $glc \circ pep \rightarrow pyr \circ glc6p$ has to be compiled into the couple of rules $glc \circ pep \rightarrow pyr$ and

$glc \circ pep \rightarrow glc6p$. However, according to the chosen semantics, this is an admissible transformation as far as the causality relation we are interested in is concerned. Technically, the chosen semantics is the bottom-up semantics that, starting from the given initial state iteratively computes all the "consequences" of it, according to the given set of rules. Informally speaking, at each step all the rules are checked, and the species caused by rules whose premises are fulfilled are considered as caused and added to semantics. For the assumptions being, the process necessarily converges to the finite set, if the model is finite, of the species produced by the network. Causality can be explicitly traced, for instance by keeping trace of all the elements that have caused a specie of interests, as well as the rules, i.e., as mentioned, the genes involved in the process. This semantics has been preferred to other backward semantics – from the species of interest to the species that cause them – because of its implementation simplicity, the generality of the solutions provided (the set of all the species caused are ready for further processing) and the easy treatment of cycles (instead of implementing a fair rule-selection discipline to guarantee termination, the semantics naturally follows the shortest non-cyclic paths for generating each specie). The tool has been developed in Sicstus Prolog[1], exploiting its features to keep the state of an in silico experiment alive between different, revised, executions of it, easily supporting subsequent "what-if" queries under updated hypothesis.

It is worth pointing out, as a consequence, that the causal map we are going to design is based on a monotonic assumption: once available, species always remain so, and then there may be limitations in dealing with the dynamical aspects of causality. For instance, cyclic behaviour or homeostatic states are difficult to treat. If from the one hand this approach offers good expressiveness and effectiveness, on the other hand, being conscious of the price in terms of quantitative aspects that in this way are left outside, we think about it as a basis for further refinements. A natural direction for extending the framework, according to the given logical interpretation of causality, is to accept forms of non-monotonic reasoning, i.e. (limited) forms of negation allowing, for instance, to reclaim the availability of a specie.

Experimental results

We exploited our toolkit over a biological model based upon the E. coli K-12 metabolic genotype proposed in [14] and [15]. This group of genes represents a subset of the whole genome of E. coli K-12 that includes genes encoding enzymes involved in energetic and biosynthetic metabolism.

[1] `http://www.sics.se/isl/sicstuswww/site/index.html`

Using our formalism we have described the metabolic network composed by the enzymes encoded by the selected gene set and the metabolites involved in the catalyzed reactions. We obtained a list of 120 causal rules having the form described above. As an example, we report below the description of the upper part of the glycolytic pathway specified according to our formalism:

$$glc6p \rightarrow fru6p \quad (1)$$

$$fru6p \circ ATP \rightarrow fru16p \circ ADP \quad (2)$$

$$fru16p \rightarrow gap \circ dhap \quad (3)$$

$$dhap \rightarrow gap \quad (4)$$

$$gap \rightarrow dhap \quad (5)$$

$$gap \circ nad \rightarrow bpg13 \circ nadh \quad (6)$$

Here the acronyms glc6p, fru6p, fru16p, gap, dhap, nad, nadh, bpg13 stand for Glucose 6-phosphate, Fructose 6-phosphate, Fructose1,6-bisphosphate, Glyceraldehyde 3-phosphate, Dihydroxyacetone phosphate, NAD^+, $NADH$ and 1,3 Bisphosphoglicerate, respectively. The rules $1, 2, 3, 4$ and 5 together, and 6 describe the reaction catalyzed by the enzymes phosphoglucose isomerase, 6-phosphofructo 1-chinase, fructose bisphosphate aldolase, triose phosphate isomerase, glyceraldehyde 3-phosphate dehydrogenase, respectively.

We have performed some in silico "what if" experiments, comparing the obtained results with the correspondent in vitro counterpart.

Mutually essentially genes We have simulated gene knock-out mimicking an homologous in vitro experiment presented in [16]. There, the authors silenced two target genes of E. coli K-12 (sucAB and sucCD) that encode for two enzymes (α-ketoglutarate dehydrogenase and succinyl-CoA synthase respectively) involved in the Krebs cycle. They found those genes "mutually essential" for the production of succinyl-CoA, i.e. sucAB and sucCD could be knocked-out individually, but not simultaneously in order to achieve Succinyl-CoA production. Succinyl-CoA is a critically important metabolite involved in several biochemical pathways leading e.g. to energy production or peptidoglycan biosynthesis (via Diaminopimelate).

To simulate this gene knock-out, we have removed the rules corresponding to the reactions catalysed by α-ketoglutarate dehydrogenase and succinyl-CoA synthase. Then we set the starting experimental conditions, including in the initial state all the metabolites that the cell is assumed to uptake from the external environment. This is

represented in the form of rules with no premises, as the following:

$$true \rightarrow glc$$

$$true \rightarrow pep$$

$$true \rightarrow o2$$

where *glc*, *pep*, $o2$ stand for α-D Glucose, Phosphoenolpyruvate and oxygen, respectively. Checking for the presence of succinyl-CoA at the end of the computation, we found that this metabolite was not produced (i.e. the correspondent fact was not deduced) only when both the target genes (i.e. the rules corresponding to the action of the encoded enzyme) were simultaneously turned off. This reflects what actually happens in vitro.

Gene knock-out and viability We pushed forward our experimentation performing other in silico gene knock-outs and comparing our results with the information contained in the "Geno Base" (http://ecoli.aist-nara.ac.jp/), a database entirely dedicated to E. coli K-12. In this database genes are classified according to various criteria among which their essentiality, i.e. their capability of causing cell death when turned off.

In our in silico knock-out experiments, we tried to test gene essentiality verifying whether or not our knock-out mutants exhibited features typically pertaining to living cells. We assumed that these characteristics could reasonably include the production of energy (ATP) and of not dispensable structural components, such as the cell wall and biomass in general.

We performed several tests, each time removing the rules corresponding to the enzyme encoded by the silenced genes and checking for the presence of the observed elements at the end of each computation. It turned out that, in all the tested cases, in silico mutants corresponding to real viable mutants did produce energy (ATP), biomass and cell wall. Furthermore, we have found that, in most of the cases, in silico mutants corresponding to real non-viable mutants did not produce energy and biomass. Nevertheless, in two cases we have obtained an in silico mutant potentially capable of producing energy and biomass, but corresponding to a non viable counterpart according to Geno Base. The presence of false negatives (the mutant is predicted viable, but actually it is not) is expected in our framework as a consequence of the abstraction and over-simplification we used in the model. This corresponds to the fact that something that has influence on viability and that has a causal explanation, it is not actually produced in live systems. This could depend, for instance, on the fact that we do not take into account some aspects related to dynamics that instead play an important role e.g. the simultaneous availability of two necessary ractants at a given instant. The preliminary results obtained up to now

are encouraging and make us confident of the reliability of our method. Nevertheless, further investigations are ongoing to systematically compare (and measure the accuracy of) our in silico predictions against the knock-out experiments reported in the Geno Base.

Conclusions

We have proposed a simple and skeletal language to describe metabolic networks, in terms of molecular entities and reaction rules that specify their interactions and implicitly code causal dependencies amongst reactions. It intends to be a sort of common basic language between biologists and computer scientists. We have then exploited the analogy between these reaction rules and logical implications, that have led us to develop a logical-based tool, able to mechanically deduce chains of causally related reactions. This makes the tool profitable for biologists that can have their intuitive description of the metabolic network easily translatable in the language used by the tool. Moreover, our methodology makes it possible to think about the model itself, by allowing to vary both the initial conditions and the rules. It is easy to program such modifications and evaluating the impact of changes in the hypotheses is quite immediate, because the tool quickly reacts to the queries (the typical answer time for a reasonable large network is about 1 second). The what-if approach satisfies the need to simulate and investigate the behaviour of a certain metabolic network, under different scenarios. In particular it allows to perform perturbative experiments which results are not trivial to predict. In fact, if the studied network is complex enough, it results unfeasible to estimate a priori the effect produced by a local perturbation on the overall network. Finally, we have applied our methodology to the metabolic network of the E. coli K-12 metabolic genotype. In general, the in silico experiments reflect the in vitro ones and have suggested interesting research directions. Even though we are at a preliminary stage, the results obtained up to now show our method not to underperform analogous ones. Noticeably it grounds on an formalism that allows efficient and straightforward implementations. This fact represents an advantage when compared, e.g. with approaches relying on graphs (see e.g. [9]) that, additionally, are more difficult to compose. Actually, using graphs, it result harder w.r.t. our method to combine the single building blocks to obtain the overall description of the system.

Our ultimate goal is that of supporting a heuristic process for searching causal explanations of metabolic phenomena, with in mind the "emphasis on hypothesis-driven research in biology" advocated in [1].

Acknowledgements

This work has been partially supported by the MIUR project Bisca.

References

1. Kitano H: Systems Biology: a brief overview. Science 2002, 295(5560):1662–1664.
2. Curti M, Degano P, Priami C, Baldari C: Causal pi-calculus for Biochemical Modelling. In Proceedings of Computational Methods in Systems Biology, LNCS 2602, Springer 2003.
3. Curti M, Degano P, Priami C, Baldari C: Modeling biochemical pathways through enhanced pi-calculus. Theoretical Computer Science 2004, 325:111–140.
4. Milner R, Parrow J, Walker D: A calculus of mobile processes (I and II). Information and Computation 1992, 100:1–77.
5. Eker S, Knapp M, Lincoln P, Laderoute K, Talcott C: Pathway Logic: Executable Models of Biological Network. In Proceedings of Fourth International Workshop on Rewriting Logic and Its Applications (WRLA'2002), ENTCS 71, Elsevier 2002.
6. Talcott C, Eker S, Knapp M, Lincoln P, Laderoute K: Pathway Logic Modeling of Protein Functional Domains in Signal Transduction. In Proceedings of Pacific Symposium on Biocomputing, 9 2004:568–580.
7. Tamaddoni-Nezhad A, Chaleil R, Kakas A, Muggleton S: Application of abductive ILP to learning metabolic network inhibition from temporal data. Machine Learning 2006, 64(1-3):209–230.
8. Papatheodorou I, Kakas AC, Sergot M: Inference of Gene Relations from Microarray Data by Abduction. In Proceedings of Logic Programming and Nonmonotonic Reasoning (LPNMR), LNCS 3662 2005:389–393.
9. Wunderlich Z, Mirny L: Using the topology of metabolic networks to predict viability of mutant strains. Biophysical Journal 2006, 91:2304–2311.
10. Cardelli L: Brane calculi-interactions of biological membranes. In Proceedings of Computational Methods in Systems Biology. Edited by Vincent V, Schachter V, Springer 2004:257–280.
11. Priami C, Regev A, Shapiro E, Silvermann W: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Theoretical Computer Science 2004, 325:141–167.
12. Regev A, Panina E, Silverman W, Cardelli L, Shapiro E: Bioambients: An abstraction for biological compartements. Theoretical Computer Science 2004, 325:141–167.
13. Apt K: From Logic Programming to Prolog. Prentice Hall International 1997.
14. Kayser A, Weber J, Henkt V, Rinas U: Metabolic flux analysis of Escherichia coli in glucose-limited continuous culture. Transactions on Computational System Biology VII 2005, 151:693–706.
15. Edwards M, Palson B: Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions. BMC Bioinformatics 2000, 1:1–11.
16. Yu B, Sung B, Lee J, Son S, Kim M, Kim S: sucAB and sucCD are mutually essential genes in Escherichia coli. FEMS Microbiology Letters 2006, 254(2).