On BIOCHAM Symbolic Computation Pipeline for Compiling Mathematical Functions into Biochemistry

François Fages, Mathieu Hemery, Sylvain Soliman

On BIOCHAM Symbolic Computation Pipeline for Compiling Mathematical Functions into Biochemistry

François Fages
Mathieu Hemery
Sylvain Soliman
Francois.Fages@inria.fr
Mathieu.Hemery@inria.fr
Sylvain.Soliman@inria.fr
Inria Saclay, EPI Lifeware
Palaiseau, Essonne, France

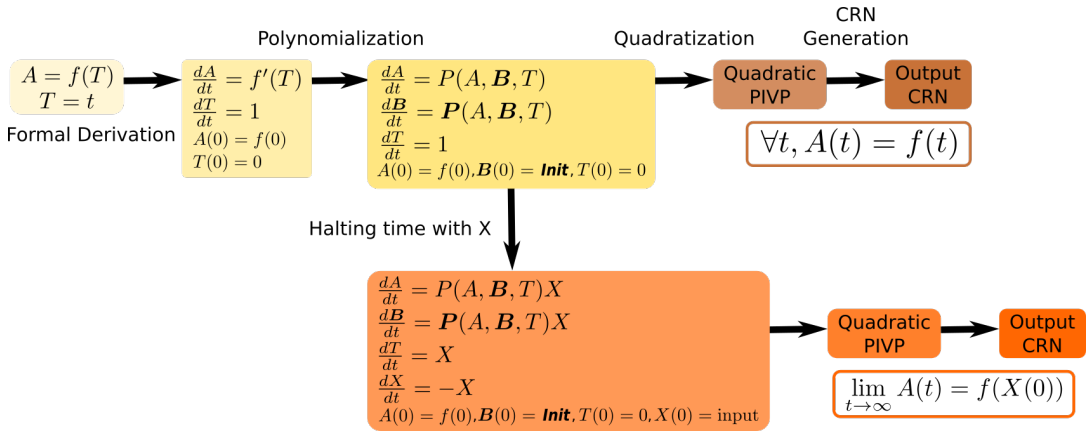

Figure 1: Symbolic computation pipeline to compile an elementary real function in a finite set of formal chemical reactions.

ABSTRACT

Chemical Reaction Networks (CRNs) are a standard formalism used in chemistry and biology to model complex molecular interaction systems. In the perspective of systems biology, they are a central tool to analyze the high-level functions of the cell in terms of their low-level molecular interactions. In the perspective of synthetic biology, they constitute a target programming language to implement in chemistry new functions either *in vitro*, in artificial vesicles, or in living cells. In this paper, we describe the CRN synthesis tool part of our CRN modeling and analysis software BIOCHAM (Biochemical Abstract Machine). This compiler transforms any elementary (resp. algebraic) real function into a formal finite CRN to compute it (resp. with absolute functional robustness), through a pipeline of symbolic computation steps, among which quadratization optimization plays a key role to restrict to elementary reactions with at most two reactants and a minimum number of molecular species.

ISSAC 2024, July 16–19, 2024, Raleigh, NC

ACM ISBN 978-1-4503-XXXX-X/24/07…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Applied computing** → **Systems biology**; **Computational biology**; • **Theory of computation** → *Models of computation.*

KEYWORDS

Analog computation, chemical computation, program synthesis, polynomialization, quadratization, stabilization, online computation, algebraic functions, elementary functions, Turing-completeness.

ACM Reference Format:
François Fages, Mathieu Hemery, and Sylvain Soliman. 2024. On BIOCHAM Symbolic Computation Pipeline for Compiling Mathematical Functions into Biochemistry. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (ISSAC 2024).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Chemical Reaction Networks (CRNs) are a standard formalism used in chemistry and biology to model complex molecular interaction systems at various levels of abstraction.

Definition 1.1. A *CRN* over a finite set of species S is a finite set of *reactions*, noted $R \xrightarrow{f} P$, where R (resp. P) is a multiset of reactant (resp. product) species, and f is a rate function over reactants, e.g. the product of the reactant concentrations by some rate constant $k \in \mathbb{R}^+$ in the case of *mass action law* kinetics.

In the perspective of systems biology, they are a central tool to analyze the high-level functions of the cell in terms of their low-level molecular interactions. In that perspective, the Systems Biology Markup Language (SBML) [22] is a common format to exchange CRN models and build CRN model repositories, such as Biomodels.net [6] which contains thousands of CRN models of a large variety of cell biochemical processes. In the perspective of synthetic biology, they constitute a target programming language to implement in chemistry new functions either *in vitro*, e.g. using DNA polymers [24], or in living cells using plasmids [11] or in artificial vesicles using proteins [9].

The mathematical theory of CRNs was introduced in the late 70's, on the one hand, by Feinberg in [15], by focusing on robust perfect adaptation (RPA) properties, Absolute Concentration Robustness (ACR) [25] and multi-stability analyses [10]; and on the other hand, by Érdi and Tóth by characterizing the set of Polynomial Ordinary Differential Equation systems (PODEs) that can be defined by CRNs with mass action law kinetics, using dual-rail encoding for negative variables [12, 13, 16, 23].

More recently, a computational theory of CRNs was investigated by formally relating their Boolean, discrete, stochastic and differential semantics in the framework of abstract interpretation [14], and by studying the computational power of CRNs under those different interpretations [7, 8, 13].

Under the continuous semantics of CRNs interpreted by ODEs, the Turing-completeness result established in [13] states that any computable real function in the sense of computable analysis, i.e. computable by a Turing machine with an arbitrary precision given in input, can be computed by a continuous CRN on a finite set of abstract molecular species, using elementary reactions with at most two reactants and mass action law kinetics. This result uses the following notion of analog computation of a non-negative real function computed by a CRN, where the result is given by the concentration of one species, y_1, and the error is controlled by the concentration of one second species, y_2:

Definition 1.2. [13] A function $f : \mathbb{R}_+ \to \mathbb{R}_+$ is CRN-computable if there exist a CRN over some molecular species $\{y_1, \ldots, y_n\}$ with differential semantics PODE $\frac{dy}{dt} = p\,(y(t))$, and a polynomial $q : \mathbb{R}_+ \to \mathbb{R}_+^n$ defining initial concentration values, such that for all $x \in \mathbb{R}_+$ there exists some (necessarily unique) function $y : \mathbb{R}_+ \to \mathbb{R}_+^n$ such that $y(0) = q(x)$, $y'(t) = p(y(t))$ and for all $t > 1$:

$$|y_1(t) - f(x)| \leq y_2(t),$$

$y_2(t) \geq 0$, $y_2(t)$ is decreasing and $\lim_{t\to\infty} y_2(t) = 0$.

Theorem 1.3. *(Turing completeness [13]) A real function is computable (resp. in polynomial time) if and only if it is CRN-computable (resp. with trajectories of polynomial length) by a CRN over molecular species for positive and negative values, using mass action law kinetics only, synthesis reactions with at most two catalysts of the form:*

$$\emptyset \to z \text{ or } x \to x + z \text{ or } x + y \to x + y + z$$

and degradation reactions by annihilation, of the form:

$$x + y \to \emptyset.$$

This result was immediately implemented in our CRN modeling, analysis and synthesis software BIOCHAM[1], the Biochemical Abstract Machine [4], with a compiler of real functions, presented as solutions of polynomial ordinary differential equations (PODE) in explicit form, into an elementary CRN over a finite set of abstract molecular species [13]. Fig. 1 summarizes the symbolic computation steps achieved to compile either a function of time or a function of some fixed input, and the two preprocessing steps of formal differentiation and polynomialization for compiling elementary real functions given in symbolic form [20]. Crucial to the size of the generated CRN is the quadratization step. We showed the NP-hardness of this optimization problem in the non-succinct matricial representation [19]. It is solved in BIOCHAM by a MAXSAT solver using a solution-preserving heuristics described here, which can compute sub-optimal solutions [2, 3].

More recently, we introduced the notions of stabilization and absolute functional robustness for the CRNs that compute a function on-line, allowing for arbitrary perturbations on the variables within the basin of attraction of the system.

Definition 1.4. [18] We say that an open CRN over a set of m+1+n species $\{X, y, Z\}$ with environment inputs X of cardinality m and distinguished output y, stabilizes the function $f : I \to \mathbb{R}_+$, with $I \subset \mathbb{R}_+^m$, over the domain $\mathcal{D} \subset \mathbb{R}_+^{m+1+n}$ if:

(1) $\forall X^0 \in I$ the restriction of the domain $\mathcal{D}$ to the slice $X = X^0$ is of plain dimension $n + 1$, and
(2) $\forall (X^0, y^0, Z^0) \in \mathcal{D}$ the Polynomial Initial Value Problem (PIVP) given by the differential semantic with constant input species $\forall t, X(t) = X^0$ and the initial conditions y^0, Z^0 is such that: $\lim_{t\to\infty} y(t) = f(X^0)$.

This definition is extended to functions from $\mathbb{R}^m$ to $\mathbb{R}$ by dual-rail encoding [12, 13, 16, 23]: for a CRN over species $\{X^+, X^-, y^+, y^-, Z\}$ we ask that $\lim_{t\to\infty}(y^+ - y^-)(t) = f(X^+ - X^-)$, for all initial conditions and constant inputs in the validity domain $\mathcal{D}$.

Let $\mathcal{F}_S$ be the set of functions stabilized by a CRN.

Definition 1.5. [18] The basin of attraction of a CRN stabilizing a function $f : I \to \mathbb{R}_+$, with $I \subset \mathbb{R}_+^m$, is the maximum domain (i.e. union of the domains) over which the CRN stabilizes f.

Theorem 1.6. *[18] The set of functions stabilized by a CRN with mass action law kinetics is the set of algebraic real functions.*

The rest of the paper presents some examples[2] and compares the main symbolic computation steps of our compiler, to generate a CRN to implement any elementary real function presented in symbolic form, or any computable real function presented as solution of a polynomial ODE system, or any algebraic function with an online stabilizing CRN.

2 EXAMPLES

Fig. 2 shows the CRN of 3 variables and 5 reactions generated by our BIOCHAM compiler for stabilizing the algebraic real function defined by the positive curve of the circle centered in $(0, 0)$.

Table 1 gives some performance figures about the complete compilation pipeline in terms of total computation time and size of

[1] Biochemical Abstract Machine software http://lifeware.inria.fr/biocham
[2] See companion notebook https://lifeware.inria.fr/wiki/Main/Software#CMSB22

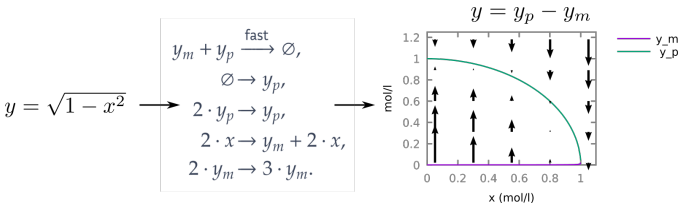

Figure 2: CRN stabilizing the circle function: $x^2 + y^2 - 1 = 0$

the synthesized CRNs on a benchmark of functions considered in [19] for the quadratization problem which is the most expensive step. It is worth noting that neither the polynomialization nor the quadratization are unique, even when imposing optimality in the number of introduced variables. The two CRNs generated for the Hill function of order 4, i.e. Hill4=$x^4/(1+x^4)$, with the two options quadratization (i.e. heuristics Alg. 3 or MAXSAT Alg. 2) are different but both have the same number of species and reactions. The Hill5 CRN is one synthetic analog of the natural MAPK signal processing network since it computes a similar input/output stiff sigmoid [21], yet using 22 species and 33 reactions. The input of the MAPK CRN is however a catalyst not consumed by the downward reactions, whereas in our CRN synthesis scheme, the input is generally consumed. This interesting case of *online* analog computation is precisely what prompted us to develop the notion of stabilization and absolute functional robustness.

	Heuristics Alg. 3			MAX-SAT Alg. 2		
Function	time (ms)	$\|S\|$	$\|CRN\|$	time (ms)	$\|S\|$	$\|CRN\|$
Hill1	80	4	5	85	3	3
Hill2	90	6	10	82	5	8
Hill3	100	6	10	115	6	12
Hill4	100	7	13	162	7	13
Hill5	110	8	16	550	7	11
Hill10	160	13	31	timeout		
Hill20	380	23	61	timeout		
Logistic	80	3	5	85	3	5
Double exp.	80	3	4	85	3	4
Gaussian	85	3	4	85	3	4
Logit	95	4	7	100	4	6

Table 1: Performance results obtained on a laptop on the benchmark of CRN design problems of [19].

3 POLYNOMIALIZATION STEP

The first step for polynomializing an ODE system consists in detecting the elements of the derivatives that are not polynomial, and their introduction as new variables. Then symbolic derivation and syntactic substitution allow us to compute the derivatives of the new variables and to modify the system of equations accordingly. In [17], we give an algorithm that terminates for any finite set F of formally differentiable functions over the reals, if $\forall f \in F, f' \in \overline{F}$ where $\overline{F}$ is the algebra of F over $\mathbb{R}$. We show that it terminates on elementary functions over the reals, with quadratic time complexity and at most a linear number of new variables. It is worth noting that the list of substitutions has to be memorized along the way, therefore handling an algebro-differential system during the execution of the algorithm, as they may reappear during the derivation step. This typically occurs when the derivation graph harbors a cycle like: cos → sin → cos. Nevertheless, a particular treatment has to be applied to the case of non-integer or negative power as they form an infinite set and may thus produce infinite chains of derivations. When adding the new variable $N = X^p$ to the system, we explicitly replace the expression X^{p-1} by N/X in the derivatives, hence the algorithm introduces the new variable $1/X$. This makes the final PODE non analytic in $X = 0$ which is linked to the fact that exponentiation apart from the polynomial case is actually not analytic in 0.

4 QUADRATIZATION STEP

The quadratization algorithm described in [5], recalled in Alg. 1, computes a quadratic form using a bounded, yet sufficient, set of monomials. We have shown that the optimal quadratization problem using that set of monomials is NP-hard.

Algorithm 1 Quadratization algorithm of Carothers et al. [5].

Input: PIVP with n variables $\{x_1, \ldots, x_n\}$, and maximum power d_j per variable.

Output: quadratic PIVP on variables v such that $v_{1,0,\ldots,0}(t) = x_1(t)$.

(1) Let A be the set of variables $v_{i_1,\ldots,i_n} = x_1^{i_1} x_2^{i_2}, \ldots, x_n^{i_n}$ for all i_j, $0 \le i_j \le d_j$, $1 \le j \le n$ satisfying $i_k > 0$ for some index k;
(2) If the output variable x_1 has a maximum power 0, add the variable $v_{1,0,\ldots} = x_1$.
(3) Compute the derivatives of the v variables as functions of the x variables;
(4) Replace the monomials in the derivatives of the v variables by monomials of the v variables with degree at most 2.

Theorem 4.1. *[19] Determining the existence of a quadratization with v monomials among those considered in Alg. 1, is NP-complete in the non-succinct matricial representation of the monomials.*

Alg. 1 can be reformulated as a MAX-SAT problem, by expressing the quadratization constraints by Boolean clauses. The maximum satisfiability problem (MAX-SAT) is a generalization of the Boolean satisfiability problem SAT, where some *soft* clauses, that can be either true or false in a solution, are added to a traditional (*hard*) SAT problem, and where the optimization problem of maximizing the number of soft clauses satisfied is considered. This leads to Alg. 2, where the number of MAX-SAT variables is $|M|$, and the number of clauses, bounded by DNF-to-CNF conversion, is $O(|M| + 2^d)$, where d is the highest product of the degrees of any monomial of m' [19].

Finally, to trade optimality for better practical performances, we use a heuristics given in Alg. 3 to restrict the set of monomials to consider and compute suboptimal solutions. We can either directly use the result of Alg. 3 or perform a MAXSAT call on top of this reduced set to try to improve it further. While not optimal this combination of heuristic and SAT-solver (`fastnSAT`) provides good results in practice.

Algorithm 2 MAX-SAT encoding of quadratization Alg. 1.

(1) For each monomial variable in A in Alg. 1, introduce a Boolean variable x_m;
(2) For each such monomial m compute its derivative m';
(3) For each monomial appearing in any m', compute all the ways to represent it as the product of 0, 1 or 2 monomials;
(4) Add one hard clause to impose the presence of the output;
(5) Add one soft clause with the negation of each other variable. The maximization will therefore try to make as few variables present as possible;
(6) Add one hard clause for each variable imposing that if it is present, the variables corresponding to the monomials of its derivative are present.

Algorithm 3 Heuristic subset of monomials.

(1) Start by determining all possible variables A and their derivatives as in Alg. 1;
(2) Initialize S to the set of output variables;
(3) Construct the set of P *problematic exponents*: those that are needed to compute the derivatives of S but cannot be reached with quadratic (or less) monomials of S;
(4) $N \leftarrow \{v \in A \setminus S | v \in P \vee \exists b \in S, vb \in P\}$, if $N = \emptyset$, $N \leftarrow A \setminus S$
(5) Determine $v^\star \in N$ that introduces as few new problematic exponents as possible.
(6) Add $v^\star$ to S and update P
(7) If $P = \emptyset$, return S otherwise go back to step (3).

5 CONCLUSION

The quadratization of a polynomial ODE is a key step in our compilation pipeline to generate an elementary CRN over a small number of formal molecular species. It is worth noting however that better solutions can be obtained by considering monomial quadratization beyond the set of monomials considered here [3], and that even better solutions exist by introducing variables for polynomials [1]. This makes of the optimal unrestricted quadratization problem an interesting open problem.

Finally, the application of our compilation pipeline to the design of concrete CRNs using real enzymes, as built for instance in artificial vesicles by a microfluidic device for diagnosis tasks [9], raises the open question of restricting our compilation pipeline to a fixed catalogue of concrete reactions at the end, involving a minimum number of enzymes and chemicals.

REFERENCES

[1] Foyez Alauddin. 2021. Quadratization of ODEs: Monomial vs. Non-Monomial. *SIAM Undergraduate Research Online* 14 (Jan. 2021). https://doi.org/10.1137/20S1360578 Sponsor: Gleb Pogudin.

[2] Andrey Bychkov, Opal Issan, Gleb Pogudin, and Boris Kramer. 2024. Exact and optimal quadratization of nonlinear finite-dimensional non-autonomous dynamical systems. *SIAM Journal on Applied Dynamical Systems* to appear (2024). arXiv:2303.10285 [cs.SC]

[3] Andrey Bychkov and Gleb Pogudin. 2021. Optimal monomial quadratization for ODE systems. In *Proceedings of the IWOCA 2021 - 32nd International Workshop on Combinatorial Algorithms*.

[4] Laurence Calzone, François Fages, and Sylvain Soliman. 2006. BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge. *Bioinformatics* 22, 14 (2006), 1805–1807. https://doi.org/10.1093/bioinformatics/btl172

[5] David C. Carothers, G. Edgar Parker, James S. Sochacki, and Paul G. Warne. 2005. Some Properties of Solutions to Polynomial Systems of Differential Equations. *Electronic Journal of Differential Equations* 2005, 40 (2005), 1–17.

[6] Vijayalakshmi Chelliah, Camille Laibe, and Nicolas Novère. 2013. BioModels Database: A Repository of Mathematical Models of Biological Processes. In *In Silico Systems Biology*, Maria Victoria Schneider (Ed.). Methods in Molecular Biology, Vol. 1021. Humana Press, 189–199. https://doi.org/10.1007/978-1-62703-450-0_10

[7] Ho-Lin Chen, David Doty, and David Soloveichik. 2012. Deterministic Function Computation with Chemical Reaction Networks. *Natural computing* 7433 (2012), 25–42. https://doi.org/10.1007/s11047-013-9393-6

[8] Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. 2009. Programmability of Chemical Reaction Networks. In *Algorithmic Bioprocesses*, Anne Condon, David Harel, Joost N. Kok, Arto Salomaa, and Erik Winfree (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 543–584. https://doi.org/10.1007/978-3-540-88869-7_27

[9] Alexis Courbet, Patrick Amar, François Fages, Eric Renard, and Franck Molina. 2018. Computer-aided biochemical programming of synthetic microreactors as diagnostic devices. *Molecular Systems Biology* 14, 4 (2018). https://doi.org/10.15252/msb.20177845

[10] Gheorghe Craciun and Martin Feinberg. 2006. Multiple equilibria in complex chemical reaction networks: II. The species-reaction graph. *SIAM J. Appl. Math.* 66, 4 (2006), 1321–1338.

[11] X. Duportet, L. Wroblewska, P. Guye, Y. Li, J. Eyquem, J. Rieders, G. Batt, and R. Weiss. 2014. A platform for rapid prototyping of synthetic gene networks in mammalian cells. *Nucleic Acids Research* 42, 21 (2014). https://doi.org/10.1093/NAR/GKU1082

[12] Péter Érdi and János Tóth. 1989. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press.

[13] François Fages, Guillaume Le Guludec, Olivier Bournez, and Amaury Pouly. 2017. Strong Turing Completeness of Continuous Chemical Reaction Networks and Compilation of Mixed Analog-Digital Programs. In *CMSB'17: Proceedings of the fiveteen international conference on Computational Methods in Systems Biology (LNCS, Vol. 10545)*. Springer-Verlag, 108–127. https://doi.org/10.1007/978-3-319-67471-1_7

[14] François Fages and Sylvain Soliman. 2008. Abstract Interpretation and Types for Systems Biology. *Theoretical Computer Science* 403, 1 (2008), 52–70. https://doi.org/10.1016/j.tcs.2008.04.024

[15] Martin Feinberg. 1977. Mathematical aspects of mass action kinetics. In *Chemical Reactor Theory: A Review*, L. Lapidus and N. R. Amundson (Eds.). Prentice-Hall, Chapter 1, 1–78.

[16] V. Hárs and J. Tóth. 1979. On the inverse problem of reaction kinetics. In *Colloquia Mathematica Societatis János Bolyai (Qualitative Theory of Differential Equations, Vol. 30)*, M. Farkas (Ed.). 363–379.

[17] Mathieu Hemery and François Fages. 2022. Algebraic Biochemistry: a Framework for Analog Online Computation in Cells. In *CMSB'22: Proceedings of the twentieth international conference on Computational Methods in Systems Biology (LNCS, Vol. 13447)*. Springer-Verlag. https://doi.org/10.1007/978-3-031-15034-0_1

[18] Mathieu Hemery and François Fages. 2024. On a model of online analog computation in the cell with absolute functional robustness: algebraic characterization, function compiler and error control. *Theoretical Computer Science* 991 (2024), 114432. https://doi.org/10.1016/j.tcs.2024.114432

[19] Mathieu Hemery, François Fages, and Sylvain Soliman. 2020. On the Complexity of Quadratization for Polynomial Differential Equations. In *CMSB'20: Proceedings of the eighteenth international conference on Computational Methods in Systems Biology (LNCS)*. Springer-Verlag. https://doi.org/10.1007/978-3-030-60327-4_7

[20] Mathieu Hemery, François Fages, and Sylvain Soliman. 2021. Compiling Elementary Mathematical Functions into Finite Chemical Reaction Networks via a Polynomialization Algorithm for ODEs. In *CMSB'21: Proceedings of the nineteenth international conference on Computational Methods in Systems Biology (LNCS, Vol. 12881)*. Springer-Verlag. https://doi.org/10.1007/978-3-030-85633-5_5

[21] Chi-Ying Huang and James E. Ferrell. 1996. Ultrasensitivity in the mitogen-activated protein kinase cascade. *PNAS* 93, 19 (Sept. 1996), 10078–10083.

[22] Michael Hucka et al. 2003. The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. *Bioinformatics* 19, 4 (2003), 524–531.

[23] K. Oishi and E. Klavins. 2011. Biomolecular implementation of linear I/O systems. *IET Systems Biology* 5, 4 (2011), 252–260.

[24] Lulu Qian, David Soloveichik, and Erik Winfree. 2011. Efficient Turing-universal computation with DNA polymers. In *Proc. DNA Computing and Molecular Programming (LNCS, Vol. 6518)*. Springer-Verlag, 123–140.

[25] Guy Shinar and Martin Feinberg. 2010. Structural Sources of Robustness in Biochemical Reaction Networks. *Science* 327, 5971 (2010), 1389–1391. https://doi.org/10.1126/science.1183372

Received 16 April 2024