Quantifying Information Flow in Chemical Reaction Networks

Ozan Kahramanoğulları[1,2]

[1] Department of Mathematics, University of Trento, Trento, Italy
[2] The Micrososft Research - University of Trento
Centre for Computational and Systems Biology, Rovereto, Italy

Abstract. We introduce an efficient algorithm for stochastic flux analysis of chemical reaction networks (CRN) that improves our previously published method for this task. The flux analysis algorithm extends Gillespie's direct method, commonly used for stochastically simulating CRNs with respect to mass action kinetics. The extension to the direct method involves only book-keeping constructs, and does not require any labeling of network species. We provide implementations, and illustrate on examples that our algorithm for stochastic flux analysis provides a means for quantifying information flow in CRNs. We conclude our discussion with a case study of the biochemical mechanism of gemcitabine, a prodrug widely used for treating various carcinomas.

Keywords: Chemical reaction networks · Stochastic simulation · Flux

1 Introduction

Chemical reaction networks (CRNs) provide a convenient representation scheme for a broad variety of models in biology and ecology. By resorting to mass action kinetics, CRNs can be simulated deterministically or stochastically. Stochastic simulations are commonly performed by using Gillespie's direct method [5], or its extensions that address a variety of concerns such as efficiency, e.g., [7], computation of rare events, e.g., [11], or portability, e.g., [2].

While it is now common practice to use deterministic and stochastic simulations interchangeably for a given CRN as well as hybrid simulations [16], these methods provide their own merits in different settings. Deterministic simulations root in a rich theory that makes available various analysis techniques, including flux analysis [15], as well as efficient numerical methods that also ease practical tasks such as model fitting by linear regression. However, differential equation simulations provide only approximations of the changes in population sizes of CRNs, as random fluctuations cannot be retrieved without introducing an additional machinery on top of deterministic methods. In this respect, because it is practically implausible to directly obtain the solution of the chemical master equation (CME) [6,13] for a not-extremely-small CRN, stochastic simulation

D. Figueiredo et al. (Eds.): AlCoB 2017, LNBI 10252, pp. 155–166, 2017.
DOI: 10.1007/978-3-319-58163-7_11

algorithms come in handy for computing stochastic trajectories of CRNs. Nevertheless, there are recent efforts that address ways for pushing the envelope by using linear approximations of CME for stochastic analysis of CRNs [1].

In previous work [10], we have presented a method for stochastic flux analysis of CRNs that is based on a consideration of stochastic simulations with CRNs as non-interleaving computations of concurrent systems [8,14]. In this approach, a simulation is considered as a reduction of a complex structure, that is, the CRN, into a simpler structure, that is, the simulation trajectory. When a simulation trajectory is read as a time series, the reaction instances are totally ordered, because the time stamps of the reaction instances specify a sequential order on them. However, when the reaction instances are considered from the point of view of their causal dependencies with respect to their production and consumption relationships with each other, the simulation trajectory takes a partial order structure rather than a sequential total order structure. In order to retrieve this otherwise lost information, the algorithm in [10] labels each species instance with a unique identifier, thereby making it possible to trace each species instance during the simulation. By tracing these identifiers, the method constructs a partial order structure of species instances. This structure is then used to quantify the causal interdependence of the reaction instances, and compressed to reveal the flux graph after a number of graph transformations.

The method described above introduces a departure from the Gillespie's direct method, as this algorithm is not designed to trace individual species instances, but rather monitor the network state as a vector of species types. Although monitoring each species does not increase the complexity of the Gillespie algorithm or hamper its correctness, it introduces an overhead due to the individual representation of species. This overhead effects the simulation efficiency, and extends the simulation time in comparison to the standard Gillespie algorithm. In some cases with tens of thousands of individuals, it also introduces a limiting factor for running simulations due to the memory required to trace individuals.

In the following, we introduce an efficient algorithm for stochastic flux analysis of CRNs in the form of a simple extension of Gillespie's direct method. In this algorithm, the fluxes of a CRN are computed during simulation by updating two arrays, the size of which are bounded by the number of reactions and species-types of the CRN. Such a mechanism of book-keeping makes it possible to monitor the network state during simulation in the form of a species-type vector as in the direct method. Consequently, the algorithm computes the flux graphs without being subject to an overhead due to monitoring of the species.

As in [10], the flux graph can be extracted for any time interval, in steady or stationary state, and it provides a causality summary of the network resources, resulting in a quantification of the information flow in the simulation. We illustrate our method on experiments with example networks. We conclude our discussion with a case study of the biochemical mechanism of gemcitabine, a prodrug widely used for treating various carcinomas. The flux graphs of this network visualize how system dynamics is affected in different metabolic regimes.

The modules, including a tool for computing flux paths, and the examples below are available for download at our website.[1]

2 Stochastic Simulation and Flux Analysis

The stochastic flux analysis of chemical reaction networks [10] is a general method on discrete event systems that can be represented as Markov chains. Such systems include those that implement mass action kinetics, which are used in systems biology for modeling a broad spectrum of phenomenon from those in molecular biology to large ecosystems. We thus here focus on chemical reaction networks (CRN) as they are studied in systems biology. Specifically, we use those that are commonly simulated by the Gillespie algorithm [5], and can be approximated by deterministic ordinary differential equation systems. The methods we discuss below, however, can be generalized to systems that are represented as discrete event systems. We first review CRNs, and stochastic flux analysis. We refer to [10] for the technical definitions and examples that are not included here.

A CRN consists of a set of reactions and an initial state. A reaction

$$m_1R_1 + \ldots + m_lR_l \xrightarrow{\rho} n_1P_1 + \ldots + n_rP_r$$

describes the species $R_1, \ldots, R_l$ that reaction consumes when it occurs, and the species $P_1, \ldots, P_r$ that it produces. The constants $m_1, \ldots, m_l$ and $n_1, \ldots, n_r$ are positive integers that denote the multiplicity of the reactants that are consumed and the products that are produced, respectively, at every instance of such a reaction. The reaction rate constant ρ is a positive real number, which determines how often a reaction occurs in a system, depending on the availability of the reactants that the reaction consumes. According to the mass action kinetics, the probability of a reaction's firing at a particular state instead of another is proportional with the multiplication of ρ and the number of possible combinations of reactants at that state. In this respect, the initial state can be safely considered as a special reaction with infinite rate, which consumes a dummy species, e.g., *Init*, which is always present at the beginning of the simulation, and is immediately consumed to produce the species that are present at time 0.

Gillespie algorithm [5] and its various extensions provide an exact method for computing the reaction occurrences of CRNs. By using this algorithm, based on continuous time Markov chains, it is possible to run stochastic simulations. Such simulations can also be approximated by ordinary differential equations. However, stochastic simulations can give rise to observations that are otherwise impossible in a deterministic setting, as stochasticity provides a means for observing random fluctuations in species numbers. As an example, consider the CRN in Fig. 1, which is a Lotka-Volterra predator-prey system [12,18].

The algorithm for stochastic flux analysis builds on the Gillespie algorithm in a way that permits the tracking of individual species as they become consumed and produced throughout the simulation. By tracking these interactions, the

[1] https://sites.google.com/site/ozankahramanogullari/software.

$$\begin{array}{l}
r_1 : \quad X \xrightarrow{100.0} \cdot \\
r_2 : \quad Y \xrightarrow{300.0} 2Y \\
r_3 : X + Y \xrightarrow{1.0} 2X \\
\mathit{init} : 100X + 100Y
\end{array}$$

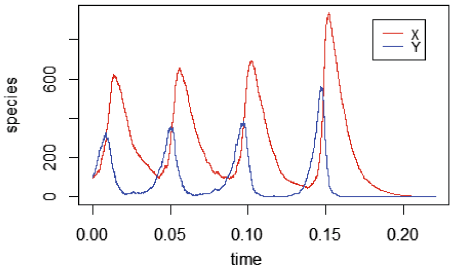


Fig. 1. A CRN model of a Lotka-Volterra predator-prey system. X represents a predator species, and Y represents a prey species. Unlike ordinary differential equation simulations, stochastic simulations as this one can capture spontaneous extinctions.

algorithm generates a quantitative log of dependencies between instances of reactions. The mechanism for this is realized by assigning a unique integer identifier to each individual species. The algorithm uses this information to incrementally construct an edge-colored graph structure by applying graph transformations. In this graph, the nodes are reactions of the CRN, and the edges are pairs of species and weights that quantify how many copies of which species flowed from which reaction to which other reaction, as exemplified in Fig. 2.

$$\begin{array}{llll}
\langle \ldots & & & \\
(r_1, \{X(1)\}, & \{\}, & 0.002); & \\
(r_1, \{X(2)\}, & \{\}, & 0.009); & \\
(r_3, \{X(6), Y(9)\}, & \{X(11), X(12)\}, & 0.013); & \rightsquigarrow \quad \ldots \quad \rightsquigarrow \\
(r_2, \{Y(8)\}, & \{Y(16), Y(17)\}, & 0.015); & \\
 & & \ldots\rangle &
\end{array}$$

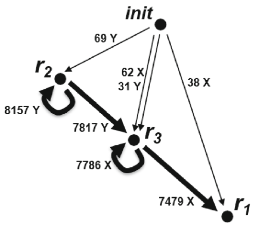


Fig. 2. Besides the time series in Fig. 1, a simulation trajectory as the one on the left is generated from the CRN during simulation as described in the text. A number of graph transformations that are applied to this structure deliver the flux graph [10].

In [10], the construction of the flux graphs is realized in a number of steps. As the first step, each species instance is assigned a unique integer identifier in the initial state. Through out the simulation, each reaction instance randomly consumes the species that match its reactants, and are randomly selected from all the possibilities. The reaction instance then introduces its products to the current state with fresh integer identifiers. Each simulation step is recorded with respect to this information in the simulation log, as exemplified in Fig. 2 for the CRN in Fig. 1. By using the unique identifiers of the species in this structure, called *reaction trajectory*, the algorithm constructs a directed acyclic graph (dag) structure, where the nodes are species instances and the edges are the reaction instances that modify these edges.

By further processing this graph, the algorithm in [10] delivers an edge-labeled directed multi-graph that reveals the independence and causality information of the transitions with respect to the flow of specific resources between reactions. Since a reaction may produce several instances of species, this structure is in general a multi-graph. This dag, highlights the production-consumption relationship between reaction instances of the simulation, and this way provides a causality history of the simulation. Flux graphs, called *flux configurations*, are then obtained by compressing these dags in order to quantify the flow of resources between the reactions within given time intervals of the simulation. More specifically, the weight of each edge specifies the number of times the species on that edge flowed from the source to the target reaction of that edge.

In order to enable the recording of the simulation trajectory as described above, the reactions act on individual instances of species, rather than types of species as it is the case in the original Gillespie algorithm. Thus, a reaction of the CRN becomes a scheme, similar to a term rewriting rule. Although this modification does not introduce an increase in the complexity of the Gillespie algorithm, neither does it hamper its correctness with respect to the mass action kinetics, it introduces an overhead due to the individual representation of species. This overhead effects the simulation efficiency, and extends the simulation time in comparison to the standard Gillespie algorithm. In some cases with tens of thousands of individuals, it also introduces a limiting factor for running simulations due to the memory required to trace individuals.

In the following, we introduce an alternative algorithm that directly constructs the flux graphs during simulation by a minimal extension of the Gillespie algorithm, and this way avoids the overhead due to the labeling of the species.

3 Refining the Stochastic Flux Analysis Algorithm

Given a CRN, the Gillepsie algorithm [5], or the SSA, is a Monte Carlo simulation procedure that faithfully selects the next reaction j and its time τ. Thus, given a CRN, an initial state, and a t_{max}, by using this algorithm a time series s can be obtained for a time interval $0 \leq t \leq t_{max}$.

Let us consider a CRN with N species $\{S_1, ..., S_N\}$, which interact through M reactions $\{R_1, ..., R_M\}$. We denote with $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$ the system state vector that represents the population of each S_i, whereby the CRN describes the time evolution of $X(t)$. The occurrence of each reaction R_j is then a discrete random event that changes the system state by $\mathbf{v}_j = \mathbf{p}_j - \mathbf{r}_j = (v_{1j}, \ldots, v_{Nj})$. The ith element v_{ij} specifies the change in X_i by one R_j reaction event, whereby p_{ij} specifies the products added to the state due to the right-hand-side of R_j, and r_{ij} specifies the reactants removed from the state due to the left-hand-side of R_j. Thus, given the system is in state $\mathbf{x} = (x_1, \ldots, x_N)$, the system jumps to state $\mathbf{x}' = \mathbf{x} + \mathbf{v}_j = \mathbf{x} + \mathbf{p}_j - \mathbf{r}_j$ as a consequence of a single R_j reaction event. The time that the next event of reaction R_j occurs is governed by function a_j, the propensity function of reaction R_j, with $a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x})$, which are updated after each simulation step according to the new state $\mathbf{x}'$.

The refined algorithm, fSSA, for computing flux configurations of CRN simulations is a conservative extension of the SSA. The steps of the algorithm that extend SSA are denoted with '$(\cdot)$' in Algorithm 1. The flux configuration is computed by updating two matrices at every simulation step. The algorithm initializes an $(M+1) \times M \times N$ matrix $\mathbf{f}$ by setting all its cells to 0 (line 4). The matrix $\mathbf{f}$ delivers the simulation fluxes at the end of the simulation, as it is updated at every simulation step. The size $M+1$ at the first dimension of $\mathbf{f}$ is due to M reactions and an additional reaction for the initial state; the size M at the second dimension is due to M reactions; the size N at the third dimension is due to N species. Then, each cell $\mathbf{f}_{\ell,j,i}$ denotes the number of species S_i that flow from R_ℓ to R_j, and the matrix $\mathbf{f}$ is output together with the time series s.

Algorithm 1. fSSA

```
Input: A CRN with N species and M reactions, initial state $\mathbf{x}_0$, and $t_{max}$.
Output: A time series $s$ and a flux matrix $\mathbf{f}$.
 1: $t \leftarrow 0$
 2: $\mathbf{x} \leftarrow \mathbf{x}_0$
 3: $s \leftarrow \langle \mathbf{x}_0 \rangle$
 4: $(\cdot)$ $\mathbf{y} \leftarrow \mathbf{x}_0$
 5: $(\cdot)$ initialize $\mathbf{f}$ such that all the cells are 0.
 6: $(\cdot)$ initialize $\mathbf{m}$ such that $m_{0,i}$ is set as in $x_i$, and all others are 0.
 7: evaluate all $a_j(\mathbf{x})$ and calculate $a_0(\mathbf{x})$
 8:     while $t \leq t_{max}$ do
 9:       $\tau \leftarrow$ a sample of exponential random variable with mean $1/a_0(\mathbf{x})$
10:       $u \leftarrow$ a sample of unit uniform random variable
11:       $\mu \leftarrow$ smallest integer satisfying $\Sigma_{i=1}^{\mu} a_i(\mathbf{x}) \geq u a_0(\mathbf{x})$
12:       $t \leftarrow t + \tau$
13:       $\mathbf{x} \leftarrow \mathbf{x} + v_\mu$
14:       $s \leftarrow s; \mathbf{x}$
15:       update $a_j(\mathbf{x})$, and recalculate $a_0(\mathbf{x})$
16: $(\cdot)$      for $i = 1$ to $N$ do
17: $(\cdot)$        for $k = 1$ to $r_{i,\mu}$ do
18: $(\cdot)$          $w \leftarrow$ a sample of unit uniform random variable
19: $(\cdot)$          $\sigma \leftarrow$ smallest integer satisfying $\Sigma_{j=1}^{\sigma} \mathbf{m}_{i,j} \geq w y_i$
20: $(\cdot)$          $\mathbf{m}_{i,\sigma} \leftarrow \mathbf{m}_{i,\sigma} - 1$
21: $(\cdot)$          $y_i \leftarrow y_i - 1$
22: $(\cdot)$          $\mathbf{f}_{\sigma,\mu,i} \leftarrow \mathbf{f}_{\sigma,\mu,i} + 1$
23: $(\cdot)$        done
24: $(\cdot)$      $\mathbf{m}_{i,\mu} \leftarrow \mathbf{m}_{i,\mu} + p_{i,\mu}$
25: $(\cdot)$      $y_i \leftarrow y_i + p_{i,\mu}$
26: $(\cdot)$      done
27:     end while
```

The second matrix that the algorithm uses for book-keeping is an $N \times (M+1)$ matrix $\mathbf{m}$, which is initialized at the beginning of the simulation, and updated at every simulation step (line 5). In $\mathbf{m}$, there are $M+1$ columns, because the first column denotes the initial state as a reaction. Thus, at time zero, the first column of $\mathbf{m}$ is initialized as the vector $\mathbf{x}_0$ and all the other cells are set to 0.

The fSSA algorithm is conservative of SSA, as it does not modify the SSA steps, and extends it with structures for logging the fluxes. The matrix $\mathbf{m}$ keeps track of the source reactions of species as they are being produced and consumed at every step. Each cell of the matrix displays a count of the species such that $\mathbf{m}_{ij}$ is the number of species S_i that had been produced by the reaction R_j, and had not been consumed by another reaction up to that point in the simulation. Thus, since each row carries the information on a certain species, each row sums up to the number of that species at the current state $\mathbf{x}$, that is, $\sum_{j=0}^{M} \mathbf{m}_{i,j} = x_i$. This information is used to sample the source of a species that can be produced by different reactions, proportional to the contribution of each reaction in producing that species (lines 15 to 22). The matrix $\mathbf{f}$ is updated accordingly (line 20). The products are then directly updated in $\mathbf{m}$ for the next simulation step (line 23).

The construction of the flux graph in Algorithm 1 introduces only a constant cost by introducing data structures that are accessed only for book-keeping, thus it is not subject to the overhead in the algorithm [10]. This is because, Algorithm 1 avoids labeling of the individual species, and this way permits the reactions to be applied on the multiplicities of instances instead of their actual occurrences.

4 Flux Paths

The FluxPath tool consists of two modules. The first module computes the flux configuration and saves this in a file. The flux configuration can be computed with the algorithm above or equivalently with the one in [10]. The former computes the flux configuration during simulation, whereas the latter first computes the simulation trajectory, which is saved to a separate file, and the flux configuration is then computed by processing this file. The second module takes the flux configuration as input and enumerates the pathways of information flow for various starting nodes and lengths of paths. The paths are computed by searching for paths in the flux configuration, which is an edge-colored weighted graph. The weight and the color, that is, the species, of each edge is kept as they are in the flux configuration during the search, and displayed in the output paths.

For an example consider the network depicted in Fig. 3 and its time series. The system implemented by this CRN is initiated in equilibrium. However, random fluctuations shift the system in a direction that favors either S2 or S5. In the simulation in Fig. 3, the dynamics results in large and small shifts at many occasions, which are visualized as fluctuations. After the time point around 550, the S2 production outweighs, which is observed as a rapid increase in S2 numbers.

By using our tool, we have analyzed the underlying dynamics with respect to the fluxes from the time point 550 to the end of the simulation; the flux graph is depicted in Fig. 4. In this graph, we observe that S4 and S6 fluxes between r_4 and r_5 have approximately the same weight, whereas S5 has a larger flux towards r_4 in comparison to S4 and S6 fluxes. Conversely, the fluxes between r_1 and r_2 weigh towards r_2. Moreover, a comparison of the S5 flux from r_3 to r_4

$$\begin{array}{llcl}
r_1 : & S1 + S2 & \xrightarrow{1.0} & S3 \\
r_2 : & S3 & \xrightarrow{1.0} & S1 + S2 \\
r_3 : & S3 & \xrightarrow{0.1} & S1 + S5 \\
r_4 : & S4 + S5 & \xrightarrow{1.0} & S6 \\
r_5 : & S6 & \xrightarrow{1.0} & S4 + S5 \\
r_6 : & S6 & \xrightarrow{0.1} & S4 + S2 \\
\multicolumn{4}{l}{init : S1 + 50S2 + S4 + 50S5}
\end{array}$$

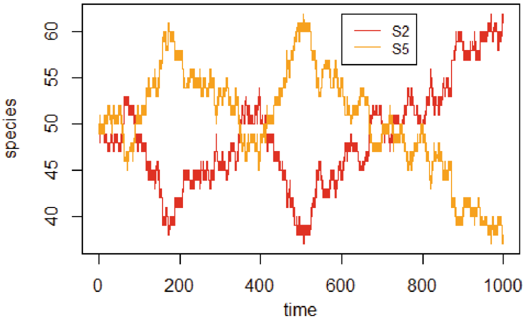


Fig. 3. A CRN of two antagonist systems; S2 and S5 compete to break the equilibrium.

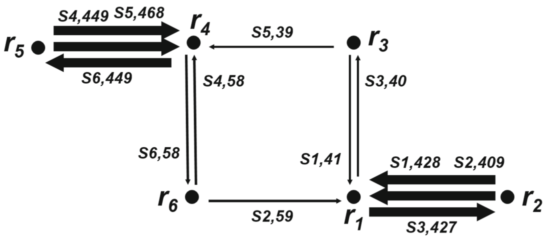


Fig. 4. The flux graph of the CRN in Fig. 3 for the time interval from 550 to 1000. The thickness of the arrows are proportional with strength of the fluxes.

and the S2 flux from r_6 to r_1 support a dynamics towards r_2. Finally, the higher turnover around r_6 in comparison to the turnover around r_3 supports the high r_2 activity that explains the shift of resources towards r_2, and the consequent excess in S2 observed in the time-series.

We have computed the paths of the flux configuration by using the FluxPath tool, available for download at our website. Among many other paths, the output flux paths depicted in Fig. 5 quantify the information flow between the S2 producing reaction r_2 and the S5 producing reaction r_5 after the time point 550.

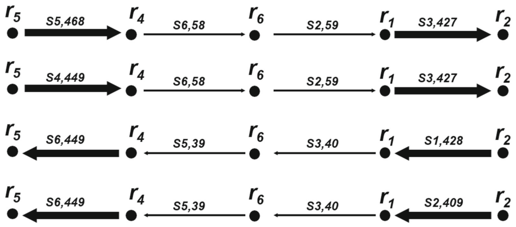


Fig. 5. The flux paths of the CRN in Fig. 3 between the S2 producing reaction r_2 and the S5 producing reaction r_5 for the time interval from 550 to 1000. The thickness of the arrows are proportional with the strength of the fluxes.

5 A Case Study: Gemcitabine

Gemcitabine (dFdC) is a prodrug, which is commonly used in the treatment of patients with non-small-cell lung cancer, pancreatic cancer, bladder cancer, and breast cancer. It is currently one of the leading therapeutic treatments for these diseases [3,4,17]. Gemcitabine exerts its clinical effect by depleting the deoxyribonucleotide pools, the building blocks of the DNA, and incorporating its triphosphate metabolite (dFdC-TP) into DNA, thereby inhibiting DNA synthesis. The incorporation of gemcitabine into DNA takes place in competition with the natural nucleotide dCTP, and this competition is an efficacy determining factor, which can be affected by various environmental and genetic conditions.

In [9], we have a given CRN model of gemcitabine biomolecular action, depicted in Fig. 6, which quantifies the the mechanisms of competition between the cascades that incorporate dCTP and dFdC-TP into the DNA. The simulations with this model identified certain mechanisms of crosstalk between these two pathways that affect the competition for incorporation. In agreement with the clinical studies dedicated to singling out mechanisms of resistance, our model associated ribonucleotide reductase (RR) and deoxycytidine kinase (dCK) activities to the efficacy of gemcitabine. Beside other mechanisms, such as transport across the plasma membrane, the inhibitory and enzymatic roles of these proteins determine efficacy depending on the availability of other metabolites.

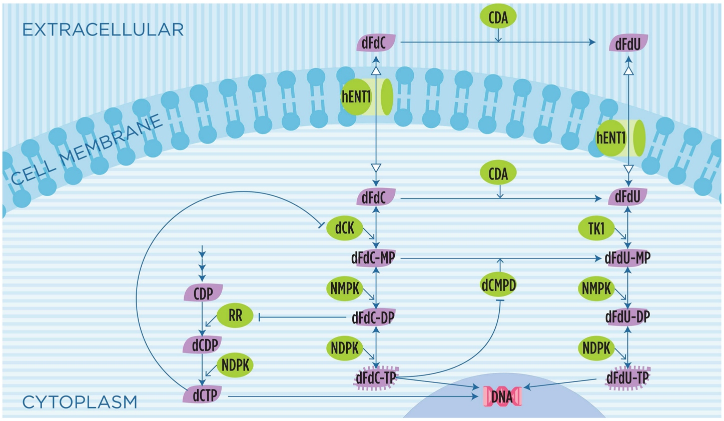
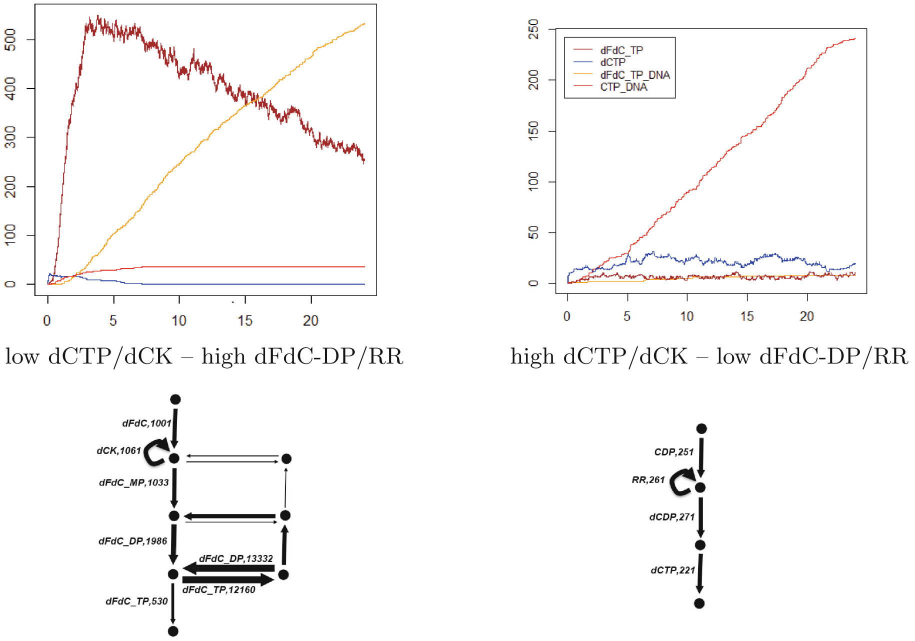


Fig. 6. The biochemical machinery of gemcitabine. Gemcitabine (dFdC and dFdU) is transported into cells by nucleoside transporters. It is then phosphorylated to its active diphosphate (dFdC-DP and dFdU-DP) and triphosphate (dFdC-TP and dFdU-TP) metabolites. Gemcitabine exerts its effect mainly by two mechanisms: while the diphosphate metabolite dFdC-DP plays an inhibitory role for the synthesis of natural nucleoside triphosphate dCTP, the triphosphate metabolite dFdC-TP competes with the dCTP for incorporation into nascent DNA chain, thereby inhibiting DNA synthesis and blocking cells in the early DNA synthesis phase. Image adopted from [9].

The efficiency of the inhibitions due to the association of dCTP with dCK and the association of dFdC-DP with RR play a key role in adjusting the efficacy. In this respect, simulations with our model have predicted a continuum of non-efficacy to high-efficacy regimes, where the levels of dFdC-TP and dCTP are coupled in a complementary manner. The complementary action, in which either dCTP or dFdC-TP make it to the DNA, is determined by the efficiency of the inhibitory associations of dCTP with dCK and dFdC-DP with RR. The extremes of this continuum are represented on one end, at the high efficacy regime, by low dCTP/dCK affinity and high dFdC-DP/RR affinity. On the other end, there is the low efficacy regime, given by high dCTP/dCK affinity and low dFdC-DP/RR affinity. Representative time series for these regimes are depicted in Figure 7.



low dCTP/dCK – high dFdC-DP/RR

high dCTP/dCK – low dFdC-DP/RR

Fig. 7. Representative time series plots and the flux pathways of the two regimes at the two ends of the efficacy spectrum of gemcitabine molecular action. The dynamics on the left is the high efficacy regime given by low dCTP/dCK affinity and high dFdC-DP/RR affinity, whereas the one on the right is the low efficacy regime given by high dCTP/dCK affinity and low dFdC-DP/RR affinity.

We have performed flux analysis by using our tool on simulations in these regimes at either ends of the spectrum to quantify the effect of the inhibitory mechanisms on information flow from outside the cell into the DNA. Flux graphs of the dominant pathways for the two cases are depicted in Fig. 7.

In the low dCTP/dCK affinity and high dFdC-DP/RR affinity regime, the increase in association of dFdC-DP and RR depletes the RR pools, and as a result of this, RR becomes unavailable to serve as an enzyme for the cascade that incorporates dCTP into the DNA. Concomitantly, the decrease in association of dCTP and dCK increases the availability of dCK to serve as enzyme in the cascade that incorporates dFdC-TP into the DNA. This results in a dominant pathway of flux depicted on the left-hand-side of Fig. 7.

At the other end of the spectrum, in the high dCTP/dCK affinity and low dFdC-DP/RR affinity regime, the complementary mechanism depletes dCK pools due to increased association of dCTP and dCK. This hampers the pathway that would otherwise incorporate the dFdC-TP into the DNA. Moreover, as a consequence of the reduction in the association of dFdC-DP and RR, more RR becomes available to serve as enzyme in the pathway that incorporates dCTP into the DNA. The resulting dynamics delivers the pathway of flux depicted on the right-hand-side of Fig. 7.

6 Discussion

We have presented a method for flux analysis in stochastic simulations of chemical reaction networks that refines our previously published method [10]. In contrast to the method in [10], the algorithm here does not require the tracking of individual species, and monitors the network state during simulation in the form of species-type vectors as in the direct method [5]. The flux graphs are then computed by instantiating and updating two arrays, the size of which are bounded by the number of reactions and species-types of the CRN. Because the algorithm is not subject to an overhead due to the number of species, it can be applied to any CRN that can be simulated with the direct method, including those with arbitrarily small species numbers. As with time series plots of stochastic simulations, simulations with greater number of events provide more convergent observations, whereas smaller number of events highlight the stochastic nature of the systems due to random fluctuations. In this respect, the method for stochastic flux analysis provides a point of view for individual simulations that is complementary to their time series considerations. The computation of the flux graphs is not restricted to steady or stationary states, and it can be performed on arbitrary time intervals as demonstrated in our examples.

Our module for computing flux paths introduces a filter that is alternative to the global view of the flux graphs, as flux paths do not have any branching. In this respect, various filters such as cut-off thresholds or filtering out certain species in flux graphs can be considered for observations on different aspects of the CRNs. Other topics of future work include implementation of an integrated modeling suit that collects features above and others, as well as investigations with a more theoretical nature, in particular, the influence of different aspects of reaction networks such as the relative contribution of structure and non-linearity to the dynamical behavior of the system, and statistical queries that can provide insights to CRN dynamics.

Acknowledgments. This work has been partially funded by the European Union's Horizon 2020 research and innovation programme under the grant agreement No 686585 - LIAR, Living Architecture.

References

1. Cardelli, L., Kwiatkowska, M., Laurenti, L.: Stochastic analysis of chemical reaction networks using linear noise approximation. Biosystems **149**, 26–33 (2016)
2. Erhard, F., Friedel, C.C., Zimmer, R.: FERN - a java framework for stochastic simulation and evaluation of reaction networks. BMC Bioinform. **9**, 356 (2008)
3. Fryer, R.A., Barlett, B., Galustian, C., Dalgleish, A.G.: Mechanisms underlying gemcitabine resistance in pancreatic cancer and sensitisation by the iMiD^{TM} lenalidomide. Anticancer Res. **31**(11), 3747–3756 (2011)
4. Funel, N., Giovannetti, E., Chiaro, M.D., Mey, V., Pollina, L.E., Nannizzi, S., Boggi, U., Ricciardi, S., Tacca, M.D., Bevilacqua, G., Mosca, F., Danesi, R., Campani, D.: Laser microdissection and primary cell cultures improve pharmacogenetic analysis in pancreatic adenocarcinoma. Lab Invest. **88**(7), 773–784 (2008)
5. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)
6. Gillespie, D.T.: A rigorous derivation of the chemical master equation. Physica A **188**, 404–425 (1992)
7. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. **115**(4), 1716 (2001)
8. Kahramanoğulları, O.: On linear logic planning and concurrency. Inf. Comput. **207**, 1229–1258 (2009)
9. Kahramanoğulları, O., Fantaccini, G., Lecca, P., Morpurgo, D., Priami, C.: Algorithmic modeling quantifies the complementary contribution of metabolic inhibitions to gemcitabine efficacy. PLoS ONE **7**(12), e50176 (2012)
10. Kahramanoğulları, O., Lynch, J.: Stochastic flux analysis of chemical reaction networks. BMC Syst. Biol. **7**, 133 (2013)
11. Kuwahara, H., Mura, I.: An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. J. Chem. Phys. **129**(16), 10B619 (2008)
12. Lotka, A.J.: Fluctuations in the abundance of a species considered mathematically. Nature **119**, 12 (1927)
13. McQuarrie, D.A.: A rigorous derivation of the chemical master equation. J. Appl. Probab. **4**, 413–478 (1967)
14. Nielsen, M., Plotkin, G., Winskel, G.: Event structures and domains, part 1. Theor. Comput. Sci. **5**(3), 223–256 (1981)
15. Okino, M.S., Mavrovouniotis, M.L.: Simplification of mathematical models of chemical reaction systems. Chem. Rev. **98**(2), 391–408 (1998)
16. Salis, H., Kaznessis, Y.N.: Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. J. Chem. Phys. **122**(5), 54103 (2005)
17. Veltkamp, S.A., Beijnen, J.H., Schellens, J.H.: Prolonged versus standard gemcitabine infusion: translation of molecular pharmacology to new treatment strategy. Oncologist **13**(3), 261–276 (2008)
18. Volterra, V.: Fluctuations in the abundance of species considered mathematically. Nature **118**, 558–560 (1926)