

# Harissa: Stochastic Simulation and Inference of Gene Regulatory Networks Based on Transcriptional Bursting

Ulysse Herbach

Université de Lorraine, CNRS, Inria, IECL, F-54000 Nancy, France  
ulysse.herbach@inria.fr

**Abstract.** Gene regulatory networks, as a powerful abstraction for describing complex biological interactions between genes through their expression products within a cell, are often regarded as virtually deterministic dynamical systems. However, this view is now being challenged by the fundamentally stochastic, ‘bursty’ nature of gene expression revealed at the single cell level. We present a Python package called Harissa which is dedicated to simulation and inference of such networks, based upon an underlying stochastic dynamical model driven by the transcriptional bursting phenomenon. As part of this tool, network inference can be interpreted as a calibration procedure for a mechanistic model: once calibrated, the model is able to capture the typical variability of single-cell data without requiring ad hoc external noise, unlike ordinary or even stochastic differential equations frequently used in this context. Therefore, Harissa can be used both as an inference tool, to reconstruct biologically relevant networks from time-course scRNA-seq data, and as a simulation tool, to generate quantitative gene expression profiles in a non-trivial way through gene interactions.

**Keywords:** Gene expression · Regulatory networks · Single-cell data

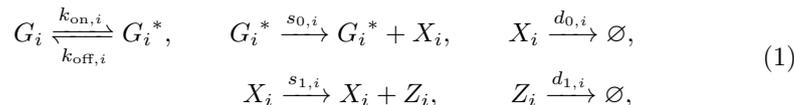
## 1 Introduction

Inferring graphs of interactions between genes has become a standard task for high-dimensional statistics, while mechanistic models describing gene expression at the molecular level have come into their own with the advent of single-cell data. Linking these two approaches seems crucial today, but the dialogue is far from obvious: statistical models often suffer from a lack of biological interpretability, and mechanistic models are known to be difficult to calibrate from real data.

Here, we present a Python package for both network simulation and inference from single-cell gene expression data (typically scRNA-seq), called Harissa (‘HARtree approximation for Inference along with a Stochastic Simulation Algorithm’). It was implemented in the context of a mechanistic approach to gene regulatory network inference from single-cell data [5] and is based upon an underlying stochastic dynamical model driven by the transcriptional bursting phenomenon. In this tool paper, we introduce briefly the main concepts behind the package, and detail its usage through some application examples.

## 2 Theory

Consider a network of  $n$  genes. Our starting point is the well-known ‘two-state model’ of gene expression [11], which corresponds to the following set of elementary chemical reactions [5] for each gene  $i \in \{1, \dots, n\}$ :



where  $G_i$ ,  $G_i^*$ ,  $X_i$  and  $Z_i$  respectively denote ‘inactive promoter’, ‘active promoter’, mRNA and protein copy numbers for gene  $i$ . These reactions describe the main two stages of gene expression, namely *transcription* (rate  $s_{0,i}$ ) and *translation* (rate  $s_{1,i}$ ), along with degradation of mRNA (rate  $d_{0,i}$ ) and protein (rate  $d_{1,i}$ ) molecules. Note that  $[G_i] + [G_i^*] = 1$  is a conserved quantity: the particularity of the two-state model is that transcription of gene  $i$  can only occur when its promoter is active, corresponding to  $[G_i^*] = 1$ .

More specifically, experimental data consistently suggest a particular regime for this model [13]:  $k_{\text{off},i} \gg k_{\text{on},i}$  and  $s_{0,i} \gg d_{0,i}$  with the ratio  $k_{\text{on},i}s_{0,i}/(k_{\text{off},i}d_{0,i})$  remaining fixed, corresponding to short active periods during which many mRNA molecules are produced. In this regime, mRNA is transcribed by ‘bursts’ of tens to hundreds of molecules. Moreover, mRNA and protein copy numbers are not conserved quantities in the model and can be reasonably described in a continuous way using standard ‘mass action’ kinetics. This leads to a hybrid dynamical model, consisting of ordinary differential equations subject to random jumps (Fig. 1) where the related quantities are denoted by  $X(t) = (X_1(t), \dots, X_n(t)) \in \mathbb{R}_+^n$  and  $Z(t) = (Z_1(t), \dots, Z_n(t)) \in \mathbb{R}_+^n$ .

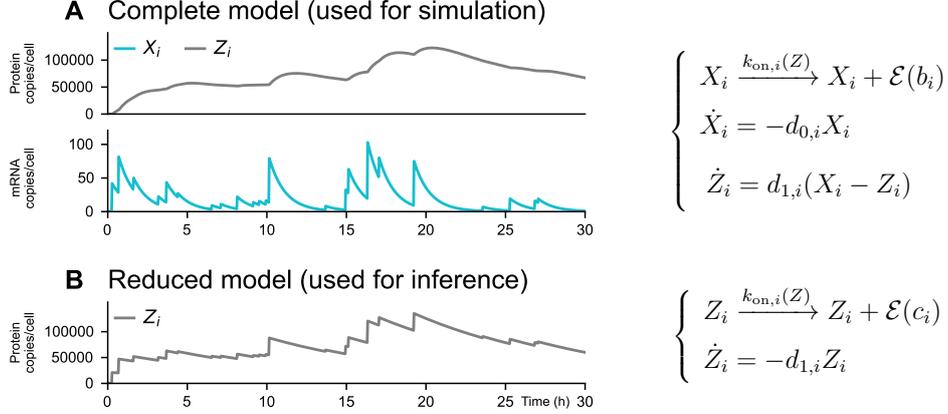
So far the genes are not interacting: the main point of our approach is to consider the burst frequency  $k_{\text{on},i}$  as a function of proteins levels  $Z_1, \dots, Z_n$  [5]. In the current version of Harissa, this function takes the following form:

$$k_{\text{on},j}(z) = \frac{k_{0,j} + k_{1,j} \exp(\beta_j + \sum_{i=1}^n \theta_{ij} z_i)}{1 + \exp(\beta_j + \sum_{i=1}^n \theta_{ij} z_i)} \quad \forall j \in \{1, \dots, n\} \quad (2)$$

with  $k_{0,j} \ll k_{1,j}$ , so that  $(\theta_{ij})_{1 \leq i, j \leq n}$  can be easily interpreted as the *network interaction matrix* while  $\beta_j$  encodes basal activity of gene  $j$  (see also Table 1). The rate  $k_{\text{off},i}$  (inverse of mean burst duration) is kept constant.

Mathematically speaking, the complete set of variables  $(X(t), Z(t))_{t \geq 0} \in \mathbb{R}_+^n \times \mathbb{R}_+^n$  forms a *piecewise-deterministic Markov process* (PDMP) that is fully characterized by its (continuous) master equation:

$$\begin{aligned} \frac{\partial}{\partial t} p(x, z, t) = & \sum_{i=1}^n \left[ d_{0,i} \frac{\partial}{\partial x_i} \{x_i p(x, z, t)\} + d_{1,i} \frac{\partial}{\partial z_i} \{(z_i - x_i) p(x, z, t)\} \right. \\ & \left. + k_{\text{on},i}(z) \left( \int_0^{x_i} p(x - h e_i, z, t) b_i e^{-b_i h} dh - p(x, z, t) \right) \right], \end{aligned} \quad (3)$$



**Fig. 1.** The stochastic dynamical model underlying Harissa. (A) Complete model with mRNA levels  $X(t)$  and protein levels  $Z(t)$ , used to perform simulations. Transcription of gene  $i$  occurs in bursts at random times with rate  $k_{\text{on},i}(Z(t))$  and the burst size follows an exponential distribution  $\mathcal{E}(b_i)$  where  $b_i$  corresponds to a scaling factor. (B) Reduced model involving only protein levels  $Z(t)$ . Without loss of generality, the parameter  $c_i$  is set to an arbitrary value as protein levels are not measured. This model serves as a basis for deriving a pseudo-likelihood whose maximization is at the core of inference.

where  $p(x, z, t)$  is the probability distribution of  $(X(t), Z(t))$ . The master equation describes the time evolution of this probability and is related to the trajectory dynamics (see Fig. 1A). Notably, each  $Z_i$  is rescaled so that parameters  $s_{1,i}$  do not appear anymore, and  $s_{0,i}$  and  $k_{\text{off},i}$  aggregate in the bursty regime into the ‘burst size’ parameter  $1/b_i = s_{0,i}/k_{\text{off},i}$  (Table 1).

This ‘complete’ model is simulated in Harissa using an efficient acceptance-rejection method, similarly to the explicit construction given in [1]. A great advantage of this method is that it is guaranteed to be exact without requiring any numerical integration, contrary to the basic algorithm [6].

We also consider a ‘protein-only’ model (Fig. 1B), which can be seen as a first-order approximation (Appendix A.1): this reduced model turns out to provide a tractable inference procedure based on analytical results (Appendix A.2).

**Table 1.** Parameters of the dynamical model underlying Harissa. Here we consider an instance `model = NetworkModel(n)` where `n` is the number of genes in the network.

	Package variable	Notation	Interpretation (gene $i$ )
<i>Degradation kinetics</i>	<code>model.d[0][i]</code>	$d_{0,i}$	mRNA degradation rate
	<code>model.d[1][i]</code>	$d_{1,i}$	protein degradation rate
<i>Bursting kinetics</i>	<code>model.a[0][i]</code>	$k_{0,i}$	minimal burst frequency
	<code>model.a[1][i]</code>	$k_{1,i}$	maximal burst frequency
	<code>model.a[2][i]</code>	$b_i$	inverse of mean burst size
<i>Network parameters</i>	<code>model.basal[i]</code>	$\beta_i$	basal activity
	<code>model.inter[i,j]</code>	$\theta_{ij}$	interaction $i \rightarrow j$

### 3 Usage

The Harissa package has two main functionalities: network inference interpreted as calibration of the stochastic dynamical model defined by (3), and data simulation from the same model (e.g., scRNA-seq counts [10] but also RT-qPCR levels [7, 12]). Besides, the package also allows for basic network visualization (directed graphs with positive or negative edge weights) as well as data binarization (using gene-specific thresholds derived from the data-calibrated dynamical model).

The first step is to create an instance of the model:

```
model = NetworkModel(n)
```

where  $n$  is the number of genes. After setting the parameter values (see Table 1), the model can be simulated using `model.simulate(time)`, where `time` is either a single time or a list of time points.

Importantly, `simulate()` does not depend on time discretization and always returns exact stochastic simulations: the resulting continuous-time trajectories are simply extracted at user-specified time points. It is also possible to consider a stimulus, represented by an additional protein  $Z_0$  that receives no feedback and verifies  $Z_0(t) = 0$  for  $t \leq 0$  and  $Z_0(t) = 1$  for  $t > 0$ . In order to reach a pre-stimulus steady state before perturbation, an optional `burnin` parameter sets the time during which the model is simulated with  $Z_0(t) = 0$ .

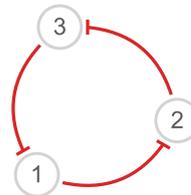
**Example: Repressilator Network.** As an example, we consider a ‘repressilator’ network made of 3 genes forming a directed cycle of negative interactions (Fig. 2).

Some critical parameters of the dynamical model are degradation rates  $d_{0,i}$  and  $d_{1,i}$ , which characterize the ‘responsiveness’ of mRNA and protein levels. Here we set  $d_{0,i}/d_{1,i} = 10$ , meaning that proteins are ten times more stable than mRNAs. This is biologically realistic, but note that this ratio is known to span a very wide range [9] so there is no single choice.

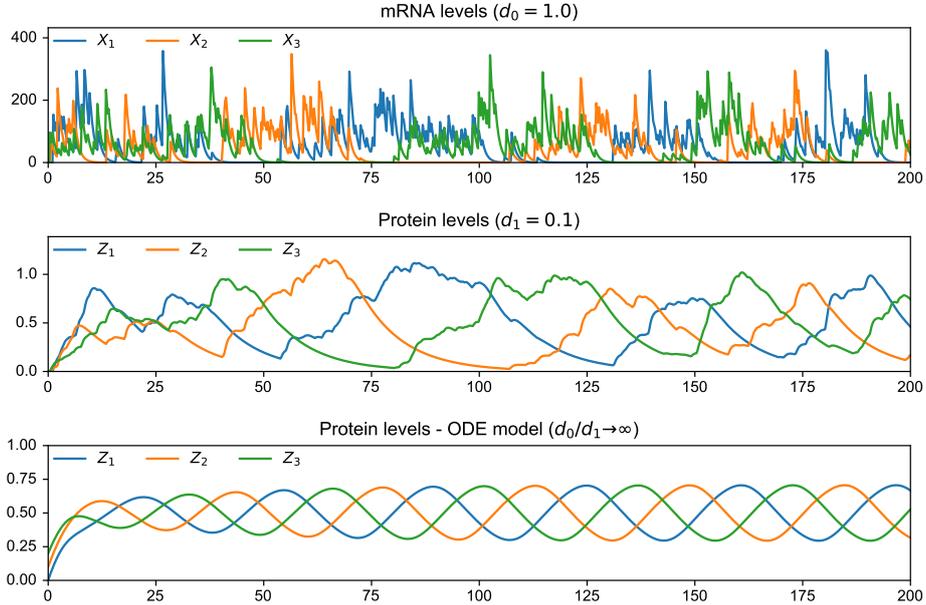
An example of simulated single-cell trajectory for this network is shown in Fig. 3. It is worth noticing that despite the strong level of stochasticity, a robust periodic pattern is already emerging.

More stable proteins—with respect to mRNA—will lead to less ‘intrinsic noise’ in the system. We increase mRNA degradation rates and burst frequencies instead, which is equivalent to a zoom-out regarding the time scale. Since mRNA and protein scales are normalized, the overall levels do not depend on degradation rates (but the *dynamics* does).

The stochastic model converges as  $d_0/d_1 \rightarrow \infty$  to a slow-fast limit [2] which turns out to be a nonlinear ODE system involving only proteins (bottom plot of Fig. 3). It appears here that due to the deterministic dynamics, the initial



**Fig. 2.** The repressilator network used in Fig. 3. This plot was made with `plot_network` from the `harissa.utils` module.

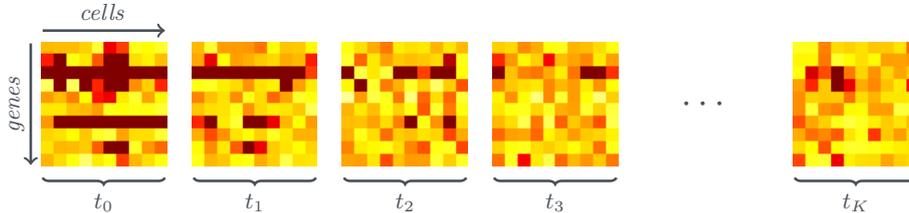


**Fig. 3.** Simulation of a repressilator network using Harissa. The first two plots show an example of single-cell trajectory of mRNA levels  $X(t)$  and protein levels  $Z(t)$  from the full stochastic model (3). The bottom plot shows a trajectory of  $Z(t)$  from the related deterministic model corresponding to the slow-fast limit (very stable proteins compared to mRNA, i.e.  $d_0/d_1 \rightarrow \infty$ ). Importantly, the periodic pattern is emerging and robust well before the limit is reached (here  $d_0/d_1 = 10$ ). The burst parameters (Table 1) are  $k_{0,i} = 0$ ,  $k_{1,i} = 2$  and  $b_i = 0.02$  for all  $i \in \{1, 2, 3\}$ , which are current default values. The network parameters are given in Appendix A.3.

protein levels need to be perturbed so as not to stay in a trivial manifold. This limit model can be simulated with `model.simulate_ode(time)` and is generally useful to gain insight into the system attractors (note however that it is a rough approximation of the stochastic model, see remark 1).

In this slow-fast limit, which is part of the rationale behind the reduced model (Appendix A.1), mRNA levels become *independent conditionally on protein levels* such that  $X_i(t) \sim \text{Gamma}(k_{\text{on},i}(Z(t))/d_{0,i}, b_i)$  for  $i = 1, 2, 3$ . This quasi-steady-state (QSS) behavior can be understood intuitively from the top plot of Fig. 3. Regarding mRNA levels, `simulate_ode` only returns the mean of the QSS distribution conditionally on protein levels (the true limit model would consist in sampling from this distribution independently for every  $t > 0$ ).

**Network Inference.** Here the main function is `model.fit()`, which takes as input a time-course single-cell dataset (Fig. 4). Inference can be performed by creating a new instance `model = NetworkModel()` without any size parameter, then loading a dataset `x` and using `model.fit(x)`. This will update all parameters



**Fig. 4.** Typical structure of time-course scRNA-seq data, as required by the inference module (here each row is a gene and each column is a cell). Each group of cells collected at the same experimental time  $t_k$  forms a *snapshot* of the biological heterogeneity at time  $t_k$ . Due to the destructive nature of the measurement process, snapshots are made of different cells. This data is therefore different from so-called ‘pseudotime’ trajectories, which attempt to reorder cells according to some smoothness hypotheses.

of the model (Table 1) except  $d_{0,i}$  and  $d_{1,i}$ , which need to be provided by the user from external data—or left to their default values, see remark 1—as they cannot be inferred without seriously compromising identifiability of other parameters. As an important feature, the updated `model` instance is ready for simulation, to assess reproducibility of the original data or to predict the outcome of network modifications. From a network inference viewpoint, the only important parameter is `model.inter` which is a non-symmetric signed weight matrix.

*Remark 1.* Degradation rates  $d_{0,i}$  and  $d_{1,i}$  are in fact not required for the *inference* part, but they are important for the *simulation* part. They are currently set by default to  $d_{0,i} = \ln(2)/9 \approx 0.077 \text{ h}^{-1}$  and  $d_{1,i} = \ln(2)/46 \approx 0.015 \text{ h}^{-1}$  as median values from thousands of genes [9]. This leads in particular to  $d_{0,i}/d_{1,i} \approx 5.11$ .

## 4 Conclusions

While the development of Harissa started a few years ago, the tool has recently become more mature and easy to use. Following early versions [5, 3], an alternative method for the inference part called Cardamom was developed in [14], which in turn influenced the current version of Harissa. The two inference methods remain complementary at this stage and may be merged into the same package in future development. They were evaluated in a recent benchmark [16], giving encouraging results and showing, most importantly, that the stochastic dynamical model (3) can indeed reproduce real time-course scRNA-seq data through gene interactions. While the simulation part is quite well optimized and can be considered as stable, the inference part is much more challenging. Future directions include a better use of the dynamical information contained in time-course transcriptional profiles, which has great potential for performance improvement.

**Code Availability.** The code of the package, a tutorial and some basic usage scripts are available at <https://github.com/ulyssheherbach/harissa>. In addition, Harissa is indexed in the Python Package Index and can be installed via `pip`.

## A Appendices

### A.1 Reduced Model

The inference procedure is based on analytical results which are not available for the two-stage ‘mRNA-protein’ model (3). On the other hand, such results exist for a one-stage ‘protein-only’ model that is a valid approximation of the former when proteins are more stable than mRNA (i.e.  $d_{0,i}/d_{1,i} \gg 1$ ). The resulting process  $(Z(t))_{t \geq 0} \in \mathbb{R}_+^n$  is also a PDMP, whose master equation can be interpreted in terms of simplified trajectories (Fig. 1B):

$$\begin{aligned} \frac{\partial}{\partial t} p(z, t) = & \sum_{i=1}^n \left[ d_{1,i} \frac{\partial}{\partial z_i} \{z_i p(z, t)\} \right. \\ & \left. + \int_0^{z_i} k_{\text{on},i}(z - he_i) p(z - he_i, t) c_i e^{-c_i h} dh - k_{\text{on},i}(z) p(z, t) \right]. \end{aligned} \quad (4)$$

Given  $Z(t)$ , mRNA levels  $X(t)$  are obtained by sampling independently for every  $i \in \{1, \dots, n\}$  and  $t > 0$  from  $X_i(t) \sim \text{Gamma}(k_{\text{on},i}(Z(t))/d_{0,i}, b_i)$ , which is the *quasi-steady-state* (QSS) distribution of the complete model [5, 6].

### A.2 Inference Algorithm

Now consider mRNA counts measured in  $m$  cells, assumed independent, along a time-course experiment following a stimulus. Each cell  $k = 1, \dots, m$  is associated with an experimental time point  $t_k$ . We introduce the following notation:

- $\mathbf{x}_k = (x_{ki}) \in \{0, 1, 2, \dots\}^n$  : mRNA counts (cell  $k$ , gene  $i$ );
- $\mathbf{z}_k = (z_{ki}) \in (0, +\infty)^n$  : latent protein levels (cell  $k$ , gene  $i$ );
- $\alpha = (\alpha_{ij}(t_k)) \in \mathbb{R}^{n \times n}$  : effective interaction  $i \rightarrow j$  at time  $t_k$ .

A stimulus is represented as gene  $i = 0$  and we therefore add parameters  $\alpha_{0j}(t_k)$  for  $j = 1, \dots, n$  and  $k = 1, \dots, m$ . We further set  $z_{k0} = 0$  if  $t_k \leq 0$  (before stimulus) and  $z_{k0} = 1$  if  $t_k > 0$  (after stimulus). Then, writing  $a_i = k_{1,i}/d_{0,i}$ , the underlying statistical model of Harissa is defined by

$$p(\mathbf{z}_k) = \prod_{i=1}^n z_{ki}^{c_i \sigma_{ki} - 1} e^{-c_i z_{ki}} \frac{c_i^{c_i \sigma_{ki}}}{\Gamma(c_i \sigma_{ki})}, \quad (5)$$

$$p(\mathbf{x}_k | \mathbf{z}_k) = \prod_{i=1}^n \frac{1}{x_{ki}!} \frac{\Gamma(a_i z_{ki} + x_{ki})}{\Gamma(a_i z_{ki})} \frac{b_i^{a_i z_{ki}}}{(b_i + 1)^{a_i z_{ki} + x_{ki}}}, \quad (6)$$

with 
$$\sigma_{ki} = \left[ 1 + \exp(-\{\beta_i + \alpha_{0i}(t_k) z_{k0} + \sum_{j=1}^n \alpha_{ji}(t_k) z_{kj}\}) \right]^{-1}. \quad (7)$$

Details of this derivation can be found in [5, 14, 4, 8]. Roughly, (5) comes from a ‘Hartree’ approximation of (4), while (6) corresponds to a Poisson distribution with random parameter sampled from the QSS distribution of  $X(t)$  given  $Z(t)$ . Note that  $p(\mathbf{z}_k)$  is in general only a pseudo-likelihood as  $\sigma_{ki}$  depends on  $\mathbf{z}_k$ .

Since the preliminary version of Harissa [5], the global inference procedure has been heavily improved using important identifiability results from [14, 15]. The final algorithm consists of three steps:

1. *Model calibration*: estimate  $a_i$  and  $b_i$  for each gene individually from (6);
2. *Bursting mode inference*: estimate the frequency mode ( $k_{0,i}$  or  $k_{1,i}$ ) for each gene in each cell (can be seen as a binarization step with specific thresholds);
3. *Network inference*: consider  $\mathbf{z}_k$  as observed from step 2 and maximize (5) with respect to  $\alpha$  after adding an appropriate penalization term [14]. Each parameter  $\theta_{ij}$  is then set to  $\alpha_{ij}(t_k)$  with  $t_k$  that maximizes  $|\alpha_{ij}(t_k)|$ .

### A.3 Repressilator Network

Considering an instance `model = NetworkModel(3)`, the repressilator network simulated in Fig. 3 is defined as follows:

```
model.d[0] = 1 # mRNA degradation rates
model.d[1] = 0.1 # Protein degradation rates
model.basal[1] = 5 # Basal activity of gene 1
model.basal[2] = 5 # Basal activity of gene 2
model.basal[3] = 5 # Basal activity of gene 3
model.inter[1,2] = -10 # Interaction 1 -> 2
model.inter[2,3] = -10 # Interaction 2 -> 3
model.inter[3,1] = -10 # Interaction 3 -> 1
```

**Acknowledgements.** The author is very grateful to Elias Ventre and Olivier Gandrillon for fruitful discussions which led to improve the Harissa package.

## References

- [1] M. Benaïm, S. Le Borgne, F. Malrieu, and P.-A. Zitt. Qualitative properties of certain piecewise deterministic Markov processes. *Annales de l'Institut Henri Poincaré - Probabilités et Statistiques*, 51(3):1040–1075, 2015. <https://doi.org/10.1214/14-AIHP619>.
- [2] A. Faggionato, D. Gabrielli, and M. Crivellari. Averaging and large deviation principles for fully-coupled piecewise deterministic Markov processes and applications to molecular motors. *Markov Processes and Related Fields*, 16(3):497–548, 2010. <https://doi.org/10.48550/arXiv.0808.1910>.
- [3] U. Herbach. *Modélisation stochastique de l'expression des gènes et inférence de réseaux de régulation*. PhD thesis, Université de Lyon, 2018.
- [4] U. Herbach. Stochastic gene expression with a multistate promoter: breaking down exact distributions. *SIAM Journal on Applied Mathematics*, 79(3):1007–1029, 2019. <https://doi.org/10.1137/18M1181006>.

- [5] U. Herbach, A. Bonnaffoux, T. Espinasse, and O. Gandrillon. Inferring gene regulatory networks from single-cell data: a mechanistic approach. *BMC Systems Biology*, 11(1):105, 2017. <https://doi.org/10.1186/s12918-017-0487-0>.
- [6] F. Malrieu. Some simple but challenging Markov processes. *Annales de la Faculté de Sciences de Toulouse*, 24(4):857–883, 2015. <https://doi.org/10.5802/afst.1468>.
- [7] A. Richard, L. Boullu, U. Herbach, A. Bonnaffoux, V. Morin, E. Vallin, A. Guillemin, N. Papili Gao, R. Gunawan, J. Cosette, O. Arnaud, J.-J. Kupiec, T. Espinasse, S. Gonin-Giraud, and O. Gandrillon. Single-cell-based analysis highlights a surge in cell-to-cell molecular variability preceding irreversible commitment in a differentiation process. *PLOS Biology*, 14(12):e1002585, 2016. <https://doi.org/10.1371/journal.pbio.1002585>.
- [8] A. Sarkar and M. Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53(6):770–777, 2021. <https://doi.org/10.1038/s41588-021-00873-4>.
- [9] B. Schwanhäusser, D. Busse, N. Li, G. Dittmar, J. Schuchhardt, J. Wolf, W. Chen, and M. Selbach. Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–342, 2011. <https://doi.org/10.1038/nature10098>.
- [10] S. Semrau, J. E. Goldmann, M. Soumillon, T. S. Mikkelsen, R. Jaenisch, and A. van Oudenaarden. Dynamics of lineage commitment revealed by single-cell transcriptomics of differentiating embryonic stem cells. *Nature Communications*, 8(1):1096, 2017. <https://doi.org/10.1038/s41467-017-01076-4>.
- [11] V. Shahrezaei and P. S. Swain. The stochastic nature of biochemical networks. *Current Opinion in Biotechnology*, 19(4):369–374, 2008. <https://doi.org/10.1016/j.copbio.2008.06.011>.
- [12] P. S. Stumpf, R. C. G. Smith, M. Lenz, A. Schuppert, F.-J. Müller, A. Babbie, T. E. Chan, M. P. H. Stumpf, C. P. Please, S. D. Howison, F. Arai, and B. D. MacArthur. Stem cell differentiation as a non-Markov stochastic process. *Cell Systems*, 5(3):268–282, 2017. <https://doi.org/10.1016/j.cels.2017.08.009>.
- [13] E. Tunnacliffe and J. R. Chubb. What is a transcriptional burst? *Trends in Genetics*, 36(4):288–297, 2020. <https://doi.org/10.1016/j.tig.2020.01.003>.
- [14] E. Ventre. Reverse engineering of a mechanistic model of gene expression using metastability and temporal dynamics. *In Silico Biology*, 14(3-4):89–113, 2021. <https://doi.org/10.3233/ISB-210226>.
- [15] E. Ventre, T. Espinasse, C.-E. Bréhier, V. Calvez, T. Lepoutre, and O. Gandrillon. Reduction of a stochastic model of gene expression: Lagrangian dynamics gives access to basins of attraction as cell types and metastability. *Journal of Mathematical Biology*, 83(5):59, 2021. <https://doi.org/10.1007/s00285-021-01684-1>.
- [16] E. Ventre, U. Herbach, T. Espinasse, G. Benoit, and O. Gandrillon. One model fits all: Combining inference and simulation of gene regulatory networks. *PLOS Computational Biology*, 19(3):e1010962, 2023. <https://doi.org/10.1371/journal.pcbi.1010962>.