UNIVERSITÉ DU LUXEMBOURG

Faculty of Science, Technology and Medicine

Master in Integrated Systems Biology

MASTER Thesis

by

María Gabriela RETAMALES BARAONA

Born on 12 July 1993 in Santiago (Chile)

Inference of Gene Regulatory Networks from Single-Cell Transcriptomics by scATA: an All-to-All Approach

UNIVERSITÉ DU LUXEMBOURG

Faculty of Science, Technology and Medicine

Master in Integrated Systems Biology

MASTER Thesis

by

María Gabriela RETAMALES BARAONA

Born on 12 July 1993 in Santiago (Chile)

Inference of Gene Regulatory Networks from Single-Cell Transcriptomics by scATA: an All-to-All Approach

Defense:	05 September 2022 in Luxembourg
Supervisor(s):	Jorge Gonçalves, PhD, Full Professor, University of Luxembourg Stefano Magni, PhD, Centre Hospitalier de Luxembourg Atte Aalto, PhD, Luxembourg Institute of Health
Jury Members:	Silvia Martina, PhD, University of Luxembourg Susan Ghaderi, PhD, KU Leuven (Belgique)

Acknowledgements

I would like to thank Professor Jorge Gonçalves for the opportunity to work in the Systems Control Group throughout the development of my thesis. I am very thankful for the interesting points of view presented and for the discussions we had, which helped me shape and improve this project. I would also like to thank Dr. Stefano Magni, from the Interventional Neuroscience group, for his constant support, his brilliant ideas, and his awesome mentoring energy. This project would have not been the same without his role as supervisor. At the same time, I want to thank Dr. Atte Aalto, for sharing his formidable scientific and technical knowledge, and for his constant feedback. Additionally, I would like to express my gratitude to all the people from the Systems Control Group at the LSCB, especially Dr. François Lamoline, for their feedback and constructive comments. I am thrilled to have been part of this amazing team. Finally, I would like to thank my family for their unconditional love and for encouraging me to follow my dreams. Without doubts this challenge would have not been possible without them.

Contents

List of Tables

List of Figures

Abstract

Gene regulatory networks (GRNs) model the controlling interactions between genes, where the expression of some genes activate or inhibit the expression of other genes. In the study of biomedical systems, a better understanding of the system can be achieved by knowing the underlying GRN in different conditions (e.g. health/disease or control/mutant). Generally, the underlying GRN of the system is not known, and it is inferred from transcriptomic data by computational methods. Single-cell transcriptomic measurements have been developed and exponentially improved over the last decade. These recent experimental techniques can measure the expression of almost each gene for most of the individual cells in a sample and have been widely used to study the heterogeneity of biological systems. However, there are not many computational methods available to infer GRNs from this type of data and the existing ones suffer from major limitations. Thus, there is a need for the development of computational approaches to infer GRNs from single-cell transcriptomics.

The aim of this thesis is to develop a simple and scalable method that can infer GRNs from single-cell transcriptomic time series data by studying pairwise regulations between genes. The presented method, named single-cell All-to-All (scATA), is based on estimating the parameters of a stochastic linear differential equation that describes the regulation between each pair of regulator and target genes, one pair at a time while ignoring other genes. The parameters are estimated by solving an optimization problem that minimizes the Wasserstein distance between the simulated distribution of the target gene and the corresponding time series data. The simulated distribution is obtained by numerically integrating a stochastic differential equation several times to obtain a distribution of the regulated gene trajectories.

The developed method was tested on synthetic data simulated from different network models with different sizes and topologies up to 10 genes, with AUROC between 0.65 and 0.91 for 5-genes networks and between 0.54 and 0.71 for 10-genes networks. The shape of the ROC curves show that, with scATA, we are able to identify a few links with high confidence. To evaluate the applicability and performance of the algorithm on experimental data, the method was applied to infer the GRN of a publicly available, single-cell transcriptomic time series data, with a publicly available GRN compiled from literature. The use of this tool can provide new insights into the regulatory mechanism inside biological systems. It can propose novel key connections between genes to be validated experimentally, that, if verified, could be useful in better understanding the underlying system and in developing targeted treatments. This thesis is as proof of concept that dynamical model-based pairwise approaches, previously used in bulk transcriptomics, can also be used for GRN inference using single-cell time series.

Preamble

Some sections of the chapter titled “Introduction” were taken and slightly modified from my previous Research Practical Report titled “Benchmark of Algorithms for Single-Cell Sequencing Simulations from a Gene Regulatory Network”. Specifically, the first three paragraphs of the section “Gene Regulatory Networks (GRNs)” and the first three paragraphs of the section “Single-cell transcriptomics”.

List of Abbreviations

ATA All-to-All

AUROC area under the receiver operating characteristic curve

CLE chemical Langevin equation

CME chemical master equation

CMP multipotent common myeloid progenitor

EML erythroid-myeloid-lymphoid

EPO erythropoietin

ERY erythroid

FN false negative

FP false positive

FPR false positive rate

GM-CSF granulocyte macrophage-colony stimulating factor

GRN gene regulatory network

IL-3 interleukin-3/n

MI mututal information

mRNA messenger RNA

MYL myeloid

NGS Next Generation Sequencing

ODE ordinary differential equation

OF objective function

PCA principal component analysis

RG regulator gene

ROC curve receiver operating characteristic curve

RT-qPCR real time quantitative polymerase chain reaction

scATA single-cell All-to-All

scRNA-Seq single-cell RNA-sequencing

SDE stochastic differential equation

TF transcription factor

TG target gene

TN true negative

TP true positive

TPR true positive rate

WD Wasserstein distance

Chapter 1

Introduction

The development of living organisms and the regulatory mechanisms governing it has been an open field of study for many centuries. In the past, it was studied by observing complete organisms, as there was no available tool to study in detail what was happening inside of them. Even with the lack of molecular experimental techniques, former scientists could account for changes in the morphological complexity of growing creatures. Nowadays, by the expansion and use of novel experimental techniques, the molecules inside the cells that regulate the developmental process have been identified and classified. These findings have led to the emergence of a new field of study aiming to understand the regulatory patterns that drive gene expression at different levels [1].

The central dogma of molecular biology describes the flow of information within the cells to go from their storage, as a DNA molecule, to their function effector, the proteins, passing through an intermediate step of RNA transcription [2]. Even though DNA is the molecule that contains the genetic information inside the cell, the expression of genes is regulated in space and time, determining the fate of different cell types and therefore their identity. Gene expression can be regulated in different parts of the transcription process: before, during, and after transcription has taken place. Transcription factors (TFs) bind to specific sequences of DNA to activate or inhibit the transcription of a specific gene. Then, the messenger RNA (mRNA) levels can be regulated by other molecules such as microRNA and long noncoding RNA. The complexity of the genetic regulatory process, and the high number of molecules involved in it have led to the description of this process as a network of interactions, also referred to as gene regulatory networks (GRNs) [1].

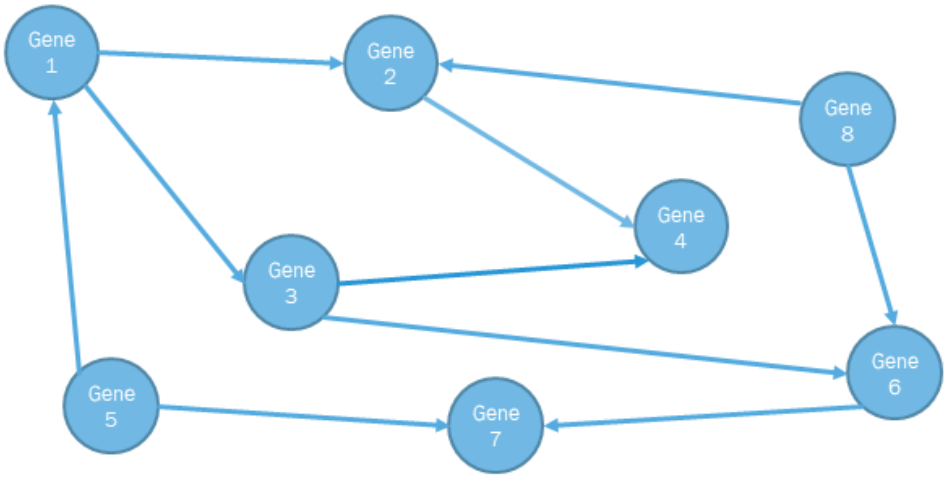


Figure 1.1: GRN illustration. Directed graph representation of an example GRN with 8 genes (nodes) and 10 regulatory interactions (edges).

1.1 Gene regulatory networks

GRN theory was developed with the objective of modelling interactions between genes and the complex control processes that occur in the cells during their cellular processes, such as development, differentiation and response to external stimuli. In a GRN, the expression of each gene in the network is associated with the expression of upstream genes regulating it, also referred to as transcriptional regulators. These transcriptional regulators can be activating or inhibiting the expression of the downstream gene. These regulatory networks organize the gene expression levels and determine which cellular functions will be occurring inside the cell [3].

From a topological perspective, GRNs consist of nodes, representing the genes, and directed edges that represent the causal interactions between them. An illustration of the graph representation of a GRNs is presented in Figure 1.1. In this figure, each edge originates from a regulator gene (RG) and ends in a target gene (TG), representing a unidirectional regulatory function [3]. As in living organisms, a particular gene can be regulated by more than one transcriptional regulator, and one transcriptional regulator can regulate multiple genes. However, most nodes have few connections, and nodes with a large number of connections, are limited. The construction of proper networks for different cell types and conditions can provide valuable insights into the different regulatory dynamics that occur inside a cell [4].

1.2 Gene regulatory network inference

To build a gene regulatory network, the biological interactions between genetic regulatory elements and their targets need to be identified. Several methodologies have been proposed to identify these connections, from computational predictions to experimentally validated interactions. Mathematical GRN inference algorithms propose interactions as new hypotheses that are then experimentally validated by using experimental techniques [2]. According to [2], there are several methods to computationally reconstruct GRNs, which differ on the type of data they use and the mathematical formalism behind them:

- Methods based on sequence-motif.

  DNA has short known and conserved sequence motifs in the promoter region of genes that can be recognized by TFs. The recognition of the motif sequence by the TF can modulate the expression of the gene by activating or inhibiting it. GRN inference methods based on sequence-motif rely on identifying these known interactions and computationally predicting new ones [2]. Some known DNA binding motif databases are MEME suite [5], miRBASE [6], and MotifMap [7]. On the other hand, some motif-based GRN prediction tools are DNAShapeR [8] and HOMER [9].

- Methods based on Chromatin Immunoprecipitation.

  Chromatin Immunoprecipitation is an experimental technique that isolates protein-DNA complexes inside the cells, allowing scientists to identify TF binding sites. It can be used to validate previously inferred regulatory relations, but also, when coupled with Next Generation Sequencing (NGS) techniques, it can propose novel interactions at a full genome scale [10].

- Methods based on gene orthology.

  These methods are based on the hypothesis that a TF-TG interaction that occurs in one species can occur in another one, and that this interaction can be identified through a phylogeny analysis. Throughout this analysis, the GRN from one species can be transferred to another species by studying the evolutionary relationship from their common ancestor [2]. Some of the computational tools that infer GRN based on gene orthology are MRTLE [11] and TargetOrtho [12].

- Methods based on open access literature.

  There are several open access databases that contain information on TF binding profiles. Some examples of these databases are TRANSFAC [13], JASPAR [14], and KEGG [15].

- Methods based on co-expression of genes.

  These methods compare the expression profile of two genes to calculate a dependency between them. Co-expression-based methods benefit from the amount of information generated with high-throughput platforms, such as NGS, and compare the expression profile of several genes by analysing different samples [2]. The typical evaluation metric is correlation, especially Pearson or Spearman correlation. Some online tools that allow to study correlations between genes across different samples and platforms are Xena Browser [16] and ALCOdb [17].

  A subclass of these methods is mututal information (MI) based algorithms, where the dependency between all gene pairs is studied and compared in order to infer the GRNs. For example, ARACNe [18] algorithm uses a MI approach to study all possible pairs of RG-TG across the whole transcriptome and identifies pairs that exhibit the same fluctuations in their expression.

  A different subclass of methods based on co-expression of genes is based on describing the biological-dynamical systems by ordinary differential equations (ODEs). The aim of these methods is to reconstruct the GRN of a system by estimating the parameters that fit an ODEs system to the transcriptomic data. ODE-based methods generally use time-series data sets and can also infer non linear relations. DyDE algorithm [19] studies time-series with a pairwise approach (all possible combinations of RG-TG) and estimates the best combination of parameters that explains how the TG is linearly regulated by the RG by solving an optimization problem. A more complex approach for GRN inference is BINGO [20], where the gene expression trajectory is modelled by a nonlinear stochastic differential equation (SDE) system, of the TGs being regulated by the other genes. BINGO takes into account the low sampling of transcriptomics time-series data sets, and builds confidence matrix of the probabilities of existence of links in the GRN.

1.3 Single-cell transcriptomics

High-throughput sequencing technologies have revolutionized the way living organisms are studied. With the emergence of NGS techniques, not only the genome can be sequenced, but also the transcriptome can be studied in detail. The transcriptome is defined as the set of all RNA transcripts, which is composed by all the genes being expressed in a specific moment on a certain organism, fluid, tissue, or cell. The huge amount of detailed information provided by transcriptomic techniques,

such as micro arrays and RNA sequencing, has allowed researchers to have a better understanding of the complexity of diverse systems. However, bulk RNA-sequencing techniques require large amounts of RNA as starting material and the information provided is only on an aggregated level. This means that even though the gene expression profile of each cell is different, the information collected in this protocol does not reflect the cellular heterogeneity of the samples. The need to understand the transcriptome of each cell individually has led to the further development of single-cell techniques [21].

In contrast to bulk transcriptome sequencing, single-cell transcriptomics provide the disaggregated expression information of each cell. In this type of experiments, before proceeding with the analytical protocol, the bulk of cells in the samples are enzymatically and/or mechanically dissociated and then each of the cells is isolated into a separated compartment. Then, the cell in each compartment is lysed, the reagents needed for measuring mRNA are added to it, and the protocol takes place. If the technique is single-cell RNA-sequencing (scRNA-Seq), after the experimental part is finished, the output reads from the next-generation sequencer are filtered for quality and mapped to a reference genome to obtain the final raw read matrix [21]. A schematic representation of the protocol can be observed in Figure 1.2. As it can be deduced from the protocols available, the cells are killed during the transcriptomic experiment, making them impossible to be studied further, and just a snapshot of the cell population is obtained at that moment [22].

The rapid evolution of single-cell transcriptomic protocols led to the development of several computational tools to help scientists decode the transcriptomic profiles for this type of data. In contrast to bulk RNA data sets, single-cell transcriptomic data is characterized by its sparsity. Due to biological variability, different cells, even though they are the same cell type, can express different genes at the same time. Furthermore, during the sequencing protocol not all the mRNA transcripts in each cell are detected, leading to a raw read matrix which has a high number of entries with a value of zero, also referred to as ‘dropouts’. Nowadays, there are more than 1000 computational analysis methods available that have been built to study gene expression from different aspects, such as quality filtering, differentially expressed genes identification, cell type assignments, clustering, and trajectory inference [23].

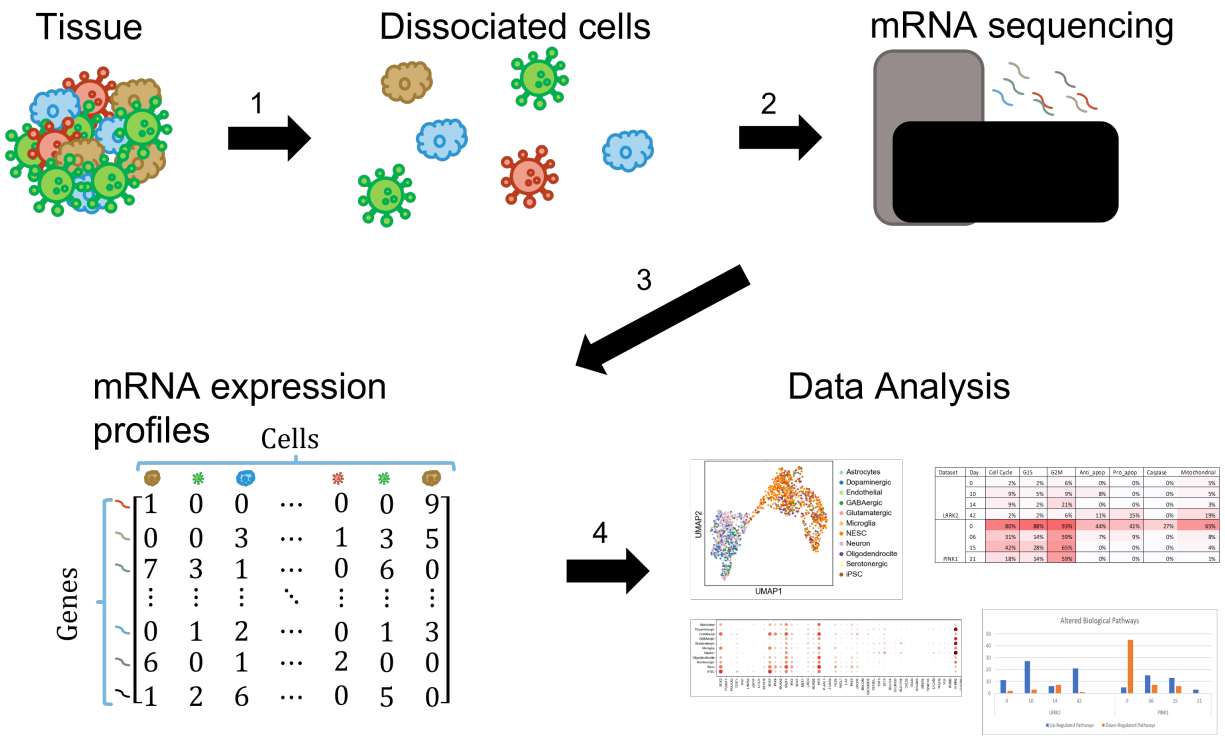


Figure 1.2: Single-cell RNA-sequencing (scRNA-Seq) workflow. The numbers in the arrows represent the following steps: Step 1) The sample is dissociated into separate cells. Step 2) The transcriptome of each cell is reverse transcribed and sequenced. Step 3) A bioinformatic pipeline is applied to obtain the raw read matrix. Step 4) The mRNA expression profiles are analyzed and compared.

1.4 Gene regulatory network inference algorithms from single-cell transcriptomic data

Algorithms that aim to infer GRNs from single-cell transcriptomics have to be able to overcome the difficulties previously mentioned for this type of data. Even though there are some algorithms and computational tools available, this is an open field of research, where algorithms are being developed [24]. Several single-cell algorithms adapt bulk transcriptomics GRN inference methods by adding a previous step to the algorithm that infers the unknown temporal order of cells. In this new step, cells are ordered based on differences in their gene expression values, in a pseudo-temporal space referred to as pseudotime [25].

The purpose of this initial pseudotime step is to explain the differences in the gene expression between cells as smooth and continuous changes [26]. Nevertheless, the development pseudo-time algorithms for single-cell transcriptomics is still in development, and the performance of the actual methods varies for different data sets and different trajectory topologies [27]. In practice, some GRN inference algorithms from single-cell data have their own implementation of pseudo-

time, while others use publicly available algorithms, such as monocle [28]. The available methods can be categorized according to their assumptions and the different inference techniques they use to identify the interactions between RGs and TGs [25]:

- Methods based on boolean models.

  Boolean models-based methods use boolean network theory to explain gene regulations. First they binarize the gene expression for each cell: 1 for expressed and 0 for non expressed genes in the cell. Subsequently, they generate an initial boolean state for the system and build an optimization method that can infer the binary functions (activation and inhibition) that can drive the system to the binarized expression [25]. Examples of these types of methods are Boolean Pseudotime [29] and BTR [30]. As it can be inferred from its name, Boolean Pseudotime first orders the cells in a pseudotemporal space.

- Methods based on differential equations.

  This class of methods for GRN inference from single-cell is similar to the one previously described on section GRN inference. To be able to apply the same kind of methods, and fit an ODE system to the studied data set, the cells have to be ordered by pseudotime. For example, SCODE [31] uses monocle to first temporarily order the cells and then fits a linear ODE system. Another example of this class of methods is SCOUP [32], which uses its own algorithm to order the cells, and then infers the GRN by modelling the continuous dynamics by stochastic diffusion.

- Methods based on gene correlation.

  This final class of single-cell GRN inference methods is based on correlating the expression of genes across the sample. They benefit from the amount of information provided by single-cell transcriptomic experiments and infer gene relationships based on different metrics such as MI, correlation distance and low-order partial correlation [25]. Some examples of these types of algorithms are Empirical Bayes [33], SINCERA [34] and SCENIC [35].

  As the previously mentioned classes, single-cell correlation based method can also benefit from the use of pseudotime by assuming that gene correlations can change during development. Therefore, some correlation-based methods calculate correlations and infer regulatory relationships only in cells that are closer in the pseudotemporal space. LEAP [36], SINGE [37] and SINCERITIES [38] are examples of this type of algorithms.

Although single-cell transcriptomic techniques provide a lot of information on the sample, there

are not many computational methods available to infer GRNs from this type of data, and the existing ones suffer from major limitations. GRN inference methods based on gene correlation identify correlation between genes, but not causality, and lack the direction of the regulation. Therefore, these methods are not suitable to study dynamical processes. On the other hand, pseudotime ordering of the cells can lead to poor accuracy of the built GRN, specially if the time trajectory topology is not linear and has several branches [25].

Overall, this creates the need for the development of methods that aim at inferring GRNs, specifically tailored to single-cell data, that can infer causal interactions and do not rely on pseudotime. Therefore, this thesis will develop a method to infer GRNs from single-cell transcriptomic data, based on dynamical models.

1.5 Biological data set to which the newly developed method for gene regulatory network inference will be applied

As mentioned before, the principal goal of developing a method to infer GRNs from different types of data sets is to better understand the biological systems these data come from. For that reason, the method developed in this thesis will be applied to study a specific biological process: the differentiation of erythroid-myeloid-lymphoid (EML) cells into erythroid (ERY) cells and myeloid (MYL) cells.

1.5.1 Stem cells

Stem cells are undifferentiated cells that can differentiate into multiple cell types and generate multiple cell lineages. These cells, with unlimited self-renewal behaviour, contribute to tissue homeostasis by generating more restricted progenitor populations that supply cells or aide in the regeneration of damaged tissue. Stem cells are classified according to their source, from embryos and fetal tissue, or from adult tissues and organs. Because of their self-renewal characteristic and their capability of differentiating into the three germ layers (ectoderm, mesoderm and endoderm), these cells have been widely used to study the cell differentiation process as well as for potential tissue regeneration and cell-based therapies for diseases such as cancer [39].

During stem cell differentiation process, also referred to as lineage commitment, the transcriptome of the cell undergoes several changes that reflect the developmental process. In their initial state, stem cells highly express genes related with pluripotency. Additionally, their state is characterised by a dynamical chromatin, which, at the same time, presents epigenetic regulation for

activation and suppression of several TF related with lineage commitment. During the subsequent differentiation process, stem cell genes are gradually shut down, making cells lose their pluripotency, and lineage specific markers increase their expression levels. Due to the heterogeneity of the cell population in which this differentiation process occurs, the transcriptomic changes of the cells allow them to be classified into their respective state commitment. Additionally, even though most of the cells undergoing the same treatment differentiate into the same cell type, not all of them do. Hence, despite the final fate of the cell being most of the times determined by its treatment, the cell fate has also been associated with stochasticity [40].

1.5.2 Blood cell progenitors

Lymphohematopoietic progenitors, EML cells, were first immortalized in 1994 and represent less than 0.01% of nucleated marrow cells. This type of cell has the ability to differentiate into lymphoid and hematopoietic cells [41]. EML cells were first purified from murine bone marrow and then infected with a retroviral vector with dominant negative retinoic acid receptor (RARα403) to establish them as a stem cell-factor dependent cell line. The cells can be cultured in stem cell-factor media, and, because of their ability to spontaneously differentiate, a typical EML suspension culture contains multipotent cells with the same characteristics as the original cell line, but can also contain cells at various differentiation stages. This cell type, because of its self-renewal and pluripotent features, has been used as a model to study cell differentiation [42].

In this context, Mojtahedi, M. and her collaborators (2016) used EML to study stability and critical states in the high dimensional system represented by the GRN underlying the differentiation of these cells, constituted of 17 genes. The EML differentiation process can be represented as a binary cell fate decision because these cells are known to differentiate into white (MYL) or red blood (ERY) cells when exposed to certain TFs (interleukin-3/n (IL-3) and granulocyte macrophage-colony stimulating factor (GM-CSF) or erythropoietin (EPO) respectively). The authors used this knowledge to understand how the initial stable state (EML cells) undergoes a critical transition to differentiate into further differentiated states [43]. EML cells have the power to differentiate into the different type of blood cells, such as ERY and MYL cells. The differentiation process of multipotent common myeloid progenitor (CMP) has been studied as a binary cell fate decision, where the genes Gata1 and sfpi1 cross-inhibit each other while they self-activate [44].

Gene regulatory network inference

During their experiments, Mojtahedi. M, *et al.* (2016) explored the EML cell differentiation process by analyzing the gene expression of 17 genes at different time points with single-cell real time quantitative polymerase chain reaction (RT-qPCR). This experimental technique measures the amount of mRNA of the desired genes. The mentioned study also presents a manually curated model of the underlying GRN that governs the fate decision of CMP cells and its following differentiation into ERY or MYL cells in its Supplementary Figure 1. [43]. An adaptation of this figure and the classification of the genes involved is available in Appendix A. Therefore, this data set will be used for testing the algorithm we will develop on real biological data with the true network known at least to some extent.

1.6 Mathematical Background

The aim of this project is to develop an algorithm for GRN inference from single-cell transcriptomic data based on dynamical models of populations. There are a three topics, from theory, that will be used throughout this project and that are important to recall. The first one is the functions used to model molecular interactions within GRNs, the second one is how to simulate the trajectory of the number of molecules of each gene in a particular cell, and the third one is how to measure the difference between two distributions as a way to evaluate how different they are. The theoretical approaches for these three concepts will be summarized below.

1.6.1 Modeling molecular interactions between genes

The complex molecular interactions (Figure 1.1) and the chemical reactions that occur inside a cell, that lead to the activation or inhibition of the genes in it, can be modelled mathematically with balance equations. Considering a closed system, these balance equations, represented as ODEs, the change of the number of mRNA molecules of a certain gene (x_i) over time is described in terms of its synthesis and its degradation:

$$\frac{dx_i}{dt} = x_i \text{ produced} - x_i \text{ degraded}. \tag{1.1}$$

For the simulation of the number of mRNA molecules of a gene in a GRN, the production rate of it is a function of the number of mRNA molecules of the genes that regulate it. On the other hand, the degradation of the mRNA molecules is usually modeled as linear, which is the natural way of biochemical molecules to be degraded. The new function for the change in mRNA molecules of

the gene would look like:

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_i, \ldots, x_N) - x_i\beta_i, \tag{1.2}$$

where $x_1, x_2, \ldots, x_i, \ldots, x_N$ are the concentrations of all N genes in the GRN, and β_i is the degradation parameter for gene x_i.

f_i from Eq. (1.2) can have different shapes that represent the regulation of the other genes in the GRN to gene i. This function must represent the activation or inhibition of the gene by others, and can be a linear or a nonlinear equation [45]. The Hill function has also been used to model these regulatory interactions, as it considers the binding affinity and can reflect cooperatively between molecules [46].

1.6.2 Single-cell simulation algorithms

Mathematical models can be used to describe the dynamics of biological systems. With the aim of modeling the heterogeneity of the cells within the samples that the resolution of single-cell techniques provides, each cell can be modeled individually. As the transcriptome of a cell is the number of RNA molecules of each gene present at a certain time, the biological states of the system can be simulated by using the same methods previously designed to model single-molecule chemical reactions [22]. Due to the small number of mRNA molecules of each gene present at the cell at a certain time, deterministic models fail to represent the stochasticity of the system properly, while models based on probability theory are more successful on this task [47]. These type of models generally assume a well-stirred, fixed volume system, where the chemical reactions occur when two or more molecules from the available chemical species collide with each other in an effective manner [48].

- The chemical master equation (CME).

  The CME, described in Eq. (1.3), is a differential equation that describes the evolution of the probability of the system being in each state, considering that the system is contained in a fixed volume, well-mixed and at fixed temperature. In a system with N chemical species and M possible reactions, the CME is defined as:

$$\dot{p}(\vec{x};t) = -p(\vec{x};t)\sum_{\mu=1}^{M} a_\mu(\vec{x}) + \sum_{\mu=1}^{M} p(\vec{x}-\nu_\mu;t)a_\mu(\vec{x}-\nu_\mu), \tag{1.3}$$

  where $p(\vec{x};t)$ is defined as the probability of $\vec{x} \in \mathbb{N}^N$ being the state vector of the system at time t. The CME describes the change of probability distribution (left-hand side of Eq. (1.3)) of the system being in a specific state $\vec{x}$ at a specific time t. $a_\mu(\vec{x})$ is the propensity function

of reaction μ to occur and ν_μ is the stoichiometric transition vector that explains how the state of the system changes when reaction μ occurs [47].

This change of probability distribution is described by the two terms on the right-hand side of Eq. (1.3). The first term, $-p(\vec{x};t)\sum_{\mu=1}^{M} a_\mu(\vec{x})$, is the probability of the system being at that specific state $\vec{x}$ and a reaction $\mu \in M$ with stoichiometric transition vector ν_μ occurring that takes the system out of that state $\vec{x}$. The second term, $\sum_{\mu=1}^{M} p(\vec{x}-\nu_\mu;t)a_\mu(\vec{x}-\nu_\mu)$, is the probability of the system not being in state $\vec{x}$ and a reaction $\mu \in M$ with stoichiometric transition vector ν_μ occurring at t that leads the system to that state. $a_\mu(\vec{x})$ is the propensity function of reaction μ to occur. The CME can also be described in terms of a matrix:

$$\dot{P}(X;t) = A \cdot P(X;t), \tag{1.4}$$

where A is the state reaction matrix and $P(X;t)$ is the complete probability density state vector at time t, that contains all the p of the possible states X ($P(X;t) = [p(x_1,t), p(x_2,t), \ldots]^T$) [47]. Because of its mathematical complexity, the CME is usually not solved analytically but it is approximated by simulation methods [22].

- Gillespie's stochastic simulation algorithm.

  The Gillespie's stochastic simulation algorithm is an algorithm from the class of Monte Carlo methods used to simulate chemical reactions when the CME cannot be solved [47]. As the CME, it considers a system with N chemical species that can interact through M chemical reactions in a well-stirred and fixed volume. This stochastic approach describes the evolution of the process in time to be led by a random-walk process, governed by the differential equation described in the CME and captures the inherent fluctuations of the system [48].

  Gillespie, D. (1977) describes the algorithm as an alternative to propagate the state of the system over time by answering in a probabilistic way two crucial questions: 1)"When will the next reaction occur?" and 2) "What kind of reaction will it be?". $P(\tau;\mu)d\tau$ is considered to be the probability density function that for a given the state $\vec{x}$ at time t, the next reaction in the volume will occur at the interval $(t+\tau, t+\tau+d\tau)$, and that this reaction will be $\mu \in M$.

  This probability can be calculated by the product of $P_0(\tau)$ and $a_\mu d\tau$. Where $P_0(\tau)$ is the probability that given the state $\vec{x}$ as time t, no reaction will occur in the interval $(t, t+\tau)$:

$$P_0(\tau) = exp(-\sum_{\mu=1}^{M} a_\mu \tau) = exp(-a_0\tau), \tag{1.5}$$

  and $a_\mu d\tau$ is the probability of reaction μ to occur in $(t+\tau, t+\tau+d\tau)$, which can be described by the product of the average probability that a particular combination of reactant molecules for μ

will react in the next $d\tau$ ($c_\mu dt$) and the number of available molecular reactants combinations for μ in the actual state (h_μ):

$$a_\mu d\tau = h_\mu c_\mu d\tau. \tag{1.6}$$

Then, $P(\tau;\mu)d\tau$ is calculated by:

$$P(\tau;\mu)d\tau = a_\mu exp(-a_0\tau), \tag{1.7}$$

[48].

Finally, the Gillespie's algorithm, described below, draws the random pair $\tau = r_1$ and $\mu = r_2$ for each iteration to calculate $P(\tau;\mu)$ and estimates the trajectory of the system. This algorithm can be run as many times as required to propagate in time the transcriptional state of many cells.

Gillespie Algorithm

From [48]

Step 0:

Input the values of the M reaction constants

c_μ for μ=1, ..., M

Input initial values for the N molecular species

$x_i(0)$ for i=1, ..., N

Set time to 0 (t = 0)

Set reaction number to 0 (n = 0)

Step 1:

Calculate propensity of each reaction

$a_\mu = h_\mu c_\mu$ for μ=1, ..., M

Calculate the total rate

$a_0 = \sum_{\mu=1}^{M} a_\mu$

Step 2:

Generate random number r_1 and r_2

Calculate $T = (1/a_0)ln(1/r_1)$

Take U so that

$\sum_{\mu=1}^{U-1} a_\mu < r_2 a_0 \leq \sum_{\mu=1}^{U} a_v$

Step 3:

Increase time by T (t=t+T)

Adjust the state vector by the reaction that occurred

$x_i = x_i + v_\mu$

Increase the reaction number by 1

Go back to Step 1.

Although it is computationally feasible to implement Gillespie's stochastic simulation algorithm to simulate the evolution of the state of the system, its computational complexity increases with the number of chemical reactions, making it unfeasible for the study of larger systems [47].

- The chemical Langevin equation (CLE).

  Gillespie, D. (2000) also demonstrated that the CME can be approximated by a system of SDEs called the CLE, described by Eq. (1.8) whenever two conditions are met: 1) "τ is small enough that the change in the state during $[t, t+\tau]$ will be so slight that none of the propensity functions changes its value 'appreciably' ", and 2) "τ is large enough that the expected number of occurrences of each reaction μ in $[t, t+\tau]$ will be much larger than 1" [49]. The CLE is defined as:

$$X_i(t+\tau) = X_i(t) + \sum_{\mu=1}^{M} \nu_{\mu i} a_\mu(\vec{X}(t))\tau + \sum_{\mu=1}^{M} \nu_{\mu i} \sqrt{a_\mu(\vec{X}(t))\tau} N_\mu(0,1). \tag{1.8}$$

  It can be noted that Eq. (1.8) uses the same notation as the CME described in Eq. (1.3), where $\vec{X}(t)$ is the state vector at time t, ν is the state change vector and $a_\mu(\vec{X})\tau$ is the propensity function, or probability that given a state vector X one reaction μ will occur in $[t, t+\tau]$. $N_\mu(0,1)$ is a normal random variable with mean 0 and variance 1 [49]. The solution of the equation for the CLE described in Eq. (1.8) can be numerically estimated by heuristic discrete time approximation algorithms, such as the Euler-Maruyama approximation described below [50].

Euler-Maruyama approximation

From [50]

Step 1:

Partition the interval [0,T] into N equal subintervals of width $\Delta t > 0$:

$0 = \tau_0 < \tau_1 = \tau_0 + \Delta t < ... < \tau_N = T$

Step 2:

Set $Y_0 = x_0$

Step 3:

Recursively define Y_n for $0 \leq n \leq N-1$ by:

$Y_{n+1} = Y_n + a(Y_n, \tau_n) * \Delta t + b(Y_n, \tau_n) * \Delta W_n$

Where $\Delta W_n = \Delta W_{\tau_{n+1}} - \Delta W_{\tau_n}$ are independent and identically distributed normal random variables with expected value zero and variance Δt.

$a(Y_n, \tau_n) = v_{i\mu} a_\mu(\vec{X}(t))$

$$b(Y_n, \tau_n) = v_{i\mu}\sqrt{a_\mu(\vec{X}(t))}$$

- Publicly available single-cell simulation tools.

  Single-cell simulation algorithms based on these mathematical principles have been published as a way of evaluating public GRN inference methods with single-cell transcriptomic data sets that have a known ground truth. GeneNetWeaver [51] and SERGIO [52] use the CLE to propagate the trajectories of cells, while Dyngen [53] uses the Gillespie's algorithm.

1.6.3 Distance between distributions

The output of single-cell transcriptomic technologies, also referred to as the raw read matrix, is a matrix where each column represents a cell, each row represents a gene, and each position in the matrix (i.e. row-column combination) represents the number of molecules of that gene detected in that cell. Therefore, the number of molecules of a gene in the population can be described in terms of a distribution [22]. Figure 1.3 is a real example of how the number of mRNA molecules in each cell for a certain gene in a data set can be represented as an histogram, which can be modeled by a distribution. Therefore, to develop an algorithm that simulates the distribution of a population, that aims at approximating the distribution of the real data, it is required to have a metric that can measure distance between both distributions (simulated and real).

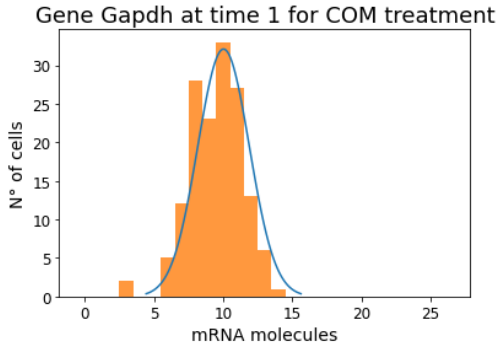


Figure 1.3: Histogram of number of cells vs mRNA molecules. Illustrative example of how the number of mRNA molecules in a population of cells for a specific gene can be displayed as a histogram and described as a distribution. In this example, the histogram represents the distribution of mRNA molecules of gene Gapdh in blood progenitor EML cells after 1 day of combined treatment to induce the differentiation of cells into MYL and ERY. The blue line over the histogram is a normal distribution with the mean and the standard deviation of the data. Figure personally generated with script 1_scAnalysis_BloodData.py with data obtained from [43].

The Wasserstein distance (WD), defined by Wasserstein in 1969 is a metric to calculate the distance between two distributions p and q for all possible pairs of random variables ξ and η from those distributions:

$$WD(p,q) = \inf \mathbb{E}\, \rho(\xi, \eta), \tag{1.9}$$

where inf is the infimum over all joint distributions, $\mathbb{E}$ is the expected value and ρ is a distance metric. If the distributions are the same, the Wasserstein distance is 0, and the more different the distributions are, the larger the WD is. Therefore, if one of the distributions is considered the truth (e.g. because it is from real or synthetic data), this metric can be used as a measure of error for the estimated distribution, quantifying how much the estimated distribution is different from the true one. [54].

In 1972, Vallender, S. demonstrated that for one-dimensional distributions, and ρ being the Euclidean distance, the WD can be calculated by the absolute difference of the cumulative density function over all the domain:

$$WD(p,q) = \int_{-\infty}^{\infty} |P(x) - Q(x)| dx. \tag{1.10}$$

This distance is calculated for two probability distributions on a line, p and q, with cumulative density functions $P(x)$ and $Q(x)$ respectively [55]. Figure 1.4 shows as an example the probability density function and the cumulative density functions for the discrete distributions A and B, where the calculated WD is the area between the two curves in the cumulative density function plot on the right. In this example, the calculated WD between distributions A and B is 2.7.

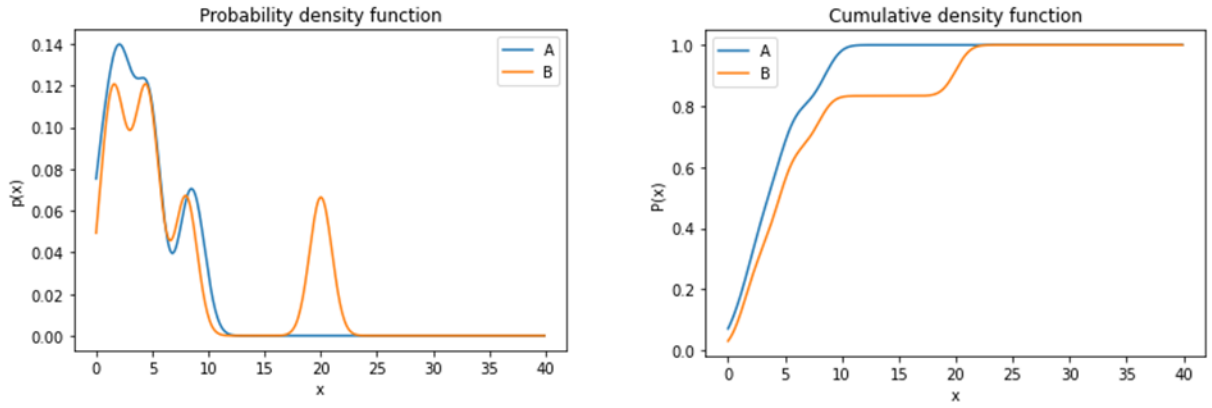


Figure 1.4: Wasserstein distance example in 1 dimension. Graphic example of how the Wasserstein distance metric is calculated between one dimensional distributions. Distribution $A = [0, 1, 2, 2, 3, 4, 5, 5, 8, 9]$ and distribution $B = [1, 2, 4, 5, 8, 20]$. The Wasserstein distance is the area between curves A and B on the cumulative density function plot (right). Personally generated with script wasserstein_distance.py

1.7 Aims of the study

The aim of this study is to develop a simple and scalable method that can infer causal regulations between genes from single-cell transcriptomic time series data. Inspired on the approach presented in [19], the method will focus on pairwise interactions, evaluating all the combinations of genes (All-to-All (ATA)). By focusing on a linear regulation of one pair of genes at a time, the method will remain computationally simple and directly parallelizable, therefore it will have the capacity to study large number of genes to detect regulations, as presented in Figure 1.5. The specific objectives of the study, which are later developed in the following chapters, are the following:

1. Determine which pairwise regulation model class will be used for the development of the algorithm. (Chapter 2)
2. Evaluate single-cell simulation algorithms to determine which will be used as mathematical formalism for the development of the algorithm and which simulation algorithm will be used to generate synthetic transcriptomic time series data to test the algorithm. (Chapter 2).
3. Select an optimization algorithm that infers the parameters of the mathematical formalism that can reproduce the gene expression distribution from the data. (Chapter 3).
4. Develop an ATA algorithm that evaluates all possible pairs of genes combinations in the optimization function and infers a GRN from single-cell transcriptomic data. (Chapter 3).
5. Test the developed algorithm on synthetically generated data sets with different number of genes and different GRN topologies. (Chapter 4).
6. Test the developed algorithm on the data set generated by [43] to evaluate the performance of the developed method with a data set that has a known ground truth, and to obtain new information on the GRN of differentiating EML cells. (Chapter 5).

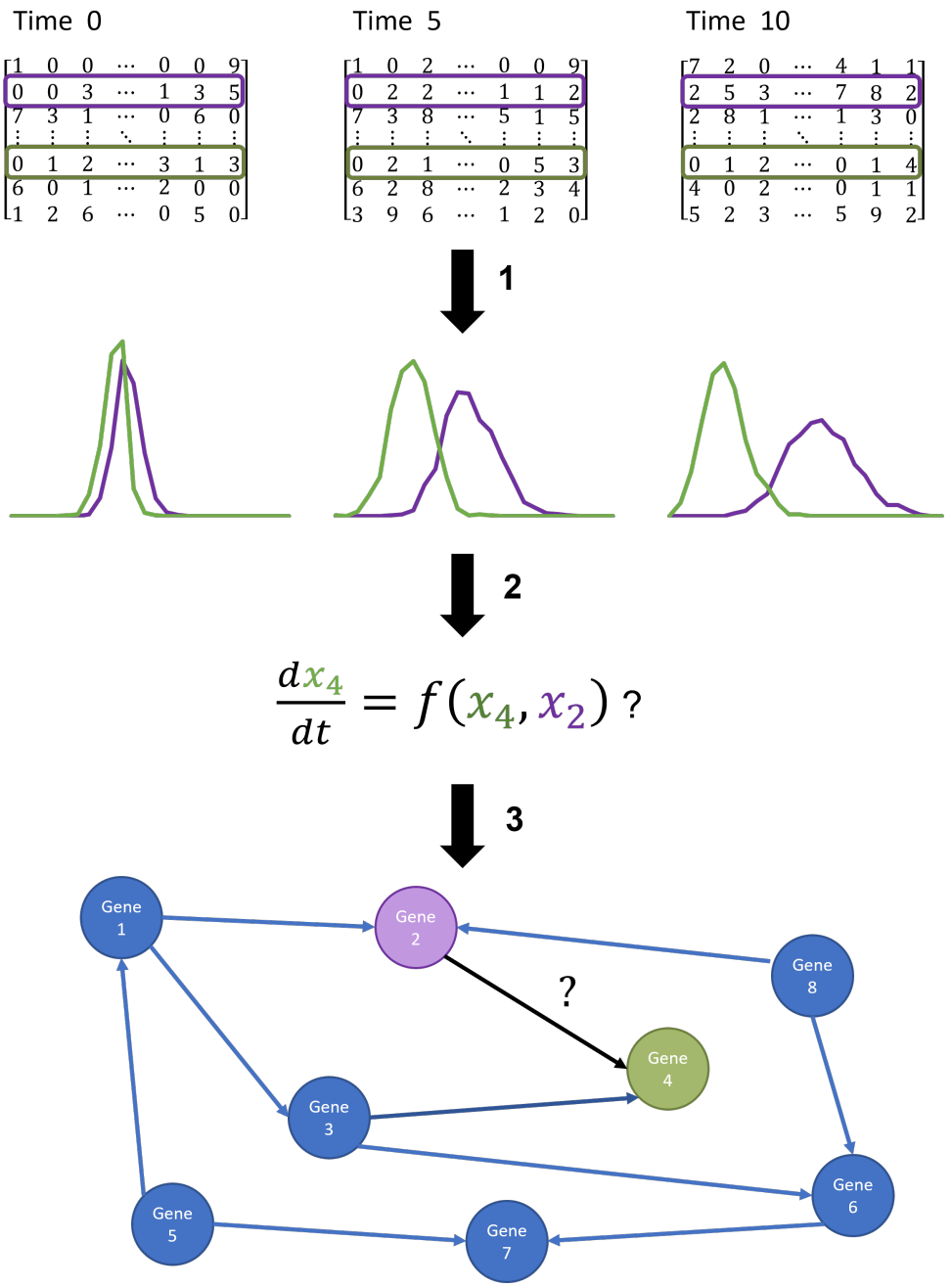


Figure 1.5: Illustration of the algorithm that will be developed in the study. Given single-cell transcriptomic time series data set, the algorithm will compare each pair of genes as regulator gene and target gene to decide if there is a regulatory relationship between them or not. In the figure, Gene2 (purple) and Gene4 (green) are compared by analyzing the profile of these genes in the data. The algorithm will study the distribution of each gene (step 1) and estimate the parameters of a function that, by simulating this equation several times, the ensemble of these simulations reproduces the data (step 2). These two steps will be repeated for every pair of genes. Finally, based on a scoring system, the algorithm will determine which links exist (step 3).

Methods & Results

Chapter 2

Simulation of synthetic single-cell transcriptomics time series data

Given the aim of building an algorithm that analyzes pairwise regulation between genes, a model class for pairwise gene regulation and a single-cell transcriptomic simulation algorithm have to be selected. Therefore, the first section of this chapter will explain the model class used, the second section will analyze single-cell simulation algorithms (CME, Gillespie's and CLE), and the final section will use the selected algorithm to simulate synthetic data for the following chapters.

2.1 Model Class

The model class was selected with the aim of building a simple model that can be tested for every pair of genes. Even though there are more complex regulation models, such as the Hill function, to maintain a low computational complexity of the final algorithm (Chapter 3), the model used in this project considers a linear regulation. Figures 2.1 and 2.2 represent the model class used throughout the project. In this model, there are two genes, a regulator (gene 1) and a target gene (gene 2). In the reactions and equations described below, the number of mRNA molecules of gene 1 and gene

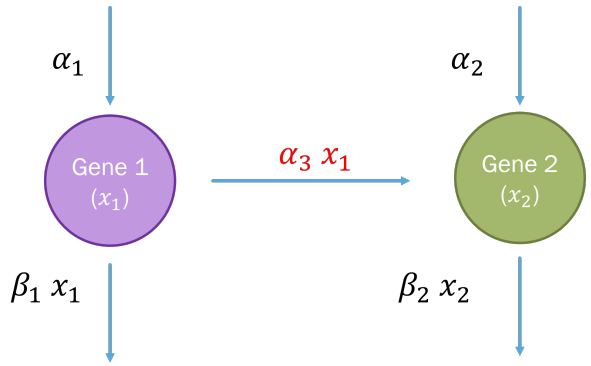


Figure 2.1: Illustration of model class used. The model class consists of 2 genes, gene 1, a RG and gene2, a TG, where gene 1 regulates gene 2. The number of mRNA molecules of these genes are represented by x_1 and x_2. Gene 1 has a production rate of α_1 and a degradation rate of $\beta_1\ x_1$. Gene 2 has a production rate of α_2 and a degradation rate of $\beta_2\ x_2$. Additionally, gene 1 regulates gene 2 by the parameter α_3 (in red), which is multiplied by the number of molecules of mRNA of the RG.

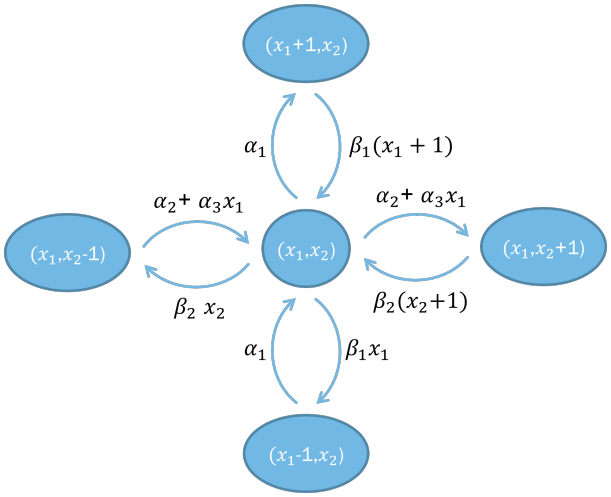


Figure 2.2: Markov process representation of model class. Only one of the five possible reactions can happen at a certain time, so the system can jump though possible discrete states where only one mRNA molecule of one gene is generated or degraded.

2 will be represented by x_1 and x_2. Gene 1 has a production rate of α_1 and a degradation rate of $\beta_1 x_1$. Gene 2 has a production rate of α_2 and a degradation rate of $\beta_2 x_2$. Additionally, gene 1 regulates gene 2 by the parameter α_3. The system can be described by reactions {1} to {5}.

$$\emptyset \xrightarrow{\alpha_1} x_1 \quad \{1\}$$

$$x_1 \xrightarrow{\beta_1 x_1} \emptyset \quad \{2\}$$

$$\emptyset \xrightarrow{\alpha_2} x_2 \quad \{3\}$$

$$\emptyset \xrightarrow{\alpha_3 x_1} x_2 \quad \{4\}$$

$$x_2 \xrightarrow{\beta_2 x_2} \emptyset \quad \{5\}$$

2.2 Simulation of transcriptomic time series data

To compare the different simulation algorithms and decide which one of them to use in the final GRN inference algorithm, the model class defined was simulated with different approaches. It was simulated by integrating the ODE system over time (without noise), by solving the CME [47],

by using the Gillespie's algorithm [48], and by numerically integrating the CLE [49] with the Euler-Maruyama algorithm [50]. Table 2.1 details the initial conditions and parameters used for simulation and comparison of each of the different simulation algorithms evaluated.

Table 2.1: Initial conditions and parameters. Number of initial molecules for each gene and values of the parameters used for simulating synthetic data for Sections 2.2.1 to 2.3.1.

Molecule	Unit of Measure	Initial Condition (IC)
x_1	Number of mRNA molecules	10
x_2	Number of mRNA molecules	10
Parameter	**Unit of Measure**	**Value**
α_1	Number of mRNA molecules / time	1
β_1	1 / time	0.05
α_2	Number of mRNA molecules / time	0.1
α_3	1 / time	0.01
β_2	1 / time	0.1

2.2.1 Simulation by integrating the ordinary differential equation system

The ODE system for the model class is described by Eqs. (2.1) and (2.2). In these equations, both genes are produced and degraded, by the production reactions {1} and {3} and degradation reactions {2} and {5} respectively. Additionally, gene 2 can also be produced by gene 1 activating gene 2, as described in reaction {4}. The ODE system is described by:

$$\frac{dx_1}{dt} = \alpha_1 - \beta_1 x_1 \tag{2.1}$$

and

$$\frac{dx_2}{dt} = \alpha_2 + \alpha_3 x_1 - \beta_2 x_2, \tag{2.2}$$

and can be solved analytically. The analytical solution for the ODE system is:

$$x_1(t) = \frac{\alpha_1}{\beta_1} - \frac{\alpha_1 - \beta_1 x_1(0)}{\beta_1} e^{-t\beta_1} \tag{2.3}$$

and

$$\begin{aligned} x_2(t) =& x_2(0)e^{-t\beta_2} \\ &+ e^{-t\beta_2}\left(\alpha_2 + \frac{\alpha_3\alpha_1}{\beta_1}\right)\frac{e^{t\beta_2} - 1}{\beta_2} \\ &- e^{-t\beta_2}\frac{\alpha_1 - \beta_1 x_1(0)}{\beta_1}\alpha_3\frac{e^{t(\beta_2-\beta_1)} - 1}{\beta_2 - \beta_1}. \end{aligned} \tag{2.4}$$

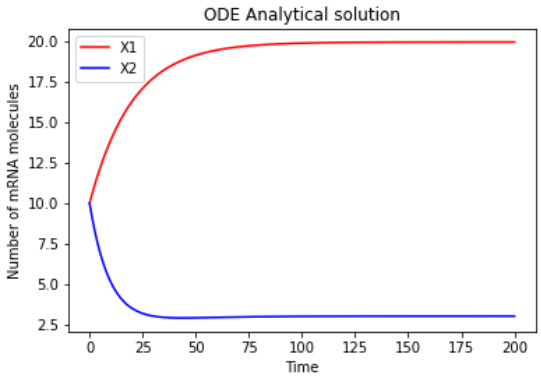


Figure 2.3: Analytical solution for the ODE system. Trajectory of the analytical solution for the evolution of the ODE system (Eqs. (2.3) and (2.4)) with the parameters from Table 2.1. Personally generated with analyticalSolSystem.py script.

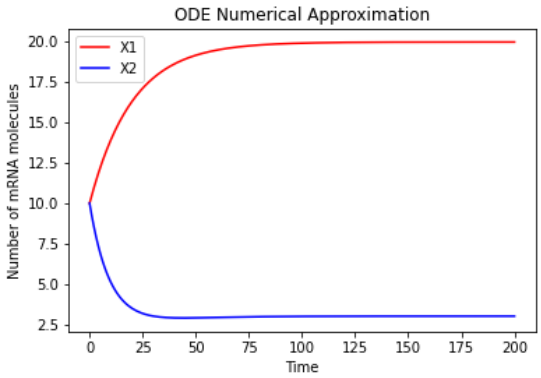


Figure 2.4: Numerical approximation for the ODE system. Trajectory of the numerical approximation for the evolution of the ODE system (Eqs. (2.1) and (2.2)) with the parameters from Table 2.1 using L-SODA numerical integrator. Personally generated with differentSimulations.py script.

Figure 2.3 shows the trajectories over time of gene 1 (x_1) and gene 2 (x_1) for the ODE system solved analytically and Figure 2.4 the same system integrated numerically.

2.2.2 Simulation by solving the chemical master equation

The CME of the system was defined by determining which state jumps could the system make from its current state and their probability (Figure 2.2). Even though the number of possible molecules at a certain time can be very big, the probability of them occurring in the system due to the set of parameters chosen is very small. Therefore, the system was truncated with a maximum of 40 mRNA molecules per gene. Appendix B details the procedure to write the model class as the system of ODEs of the CME defined in Eq. (1.3). It can be seen in that procedure that all the combinations of x_1 and x_2 have the same probability structure, except the conditions where x_1 and/or x_2 are 0 or N. In these states, the equations needed to be truncated.

To simulate the model class of five reactions, the CME was programmed so that every equation was defined in a loop, and then defining all the border reactions (x_1 or x_2 being 0 or 40). Then, the system of 1600 ODEs was integrated numerically. The result of this numerical integration was the probability distribution of the system being in each state defined by the number of molecules of gene 1 and gene 2. Figure 2.5 presents the state probability at different time points.

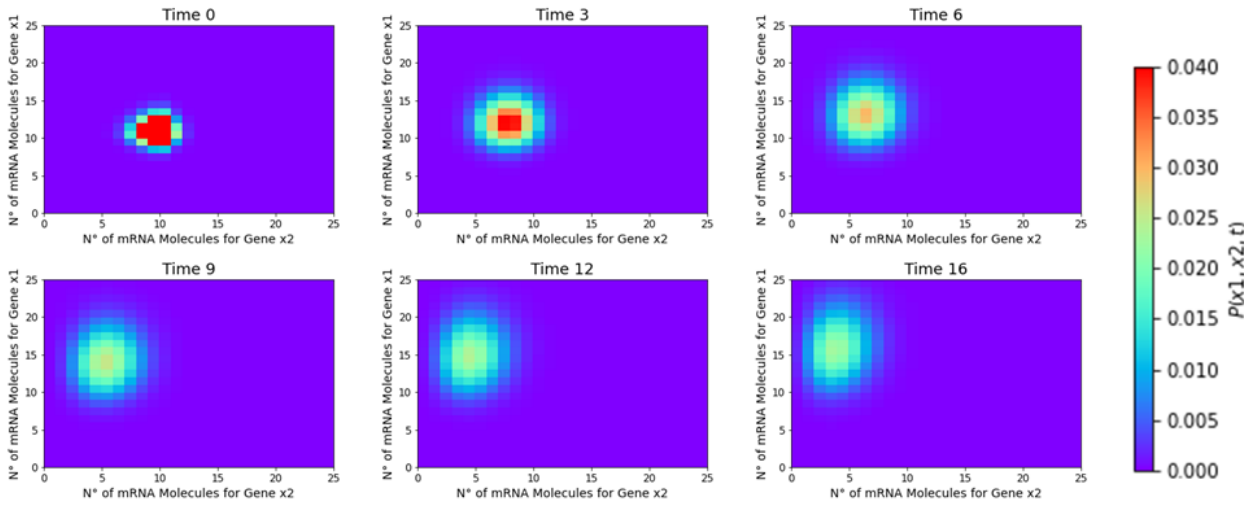


Figure 2.5: States probability distribution for different time points when numerically integrating the CME system. The model class (reactions {1} to {5)} described by the CME in Eq. (1.3) was numerically integrated, with the parameters from Table 2.1. Each figure represents the probability of the system to be in state x_1 and x_2 by the color scale on the right. Personally generated with differentSimulations.py script.

2.2.3 Simulation with Gillespie's algorithm

As described in Section 1.6.2, this algorithm can be used to numerically approximate the CME. Because it is based on trajectory simulations, and not on solving the ODE system, its computational time is lower. Therefore, the Gillespie Algorithm was used to simulate the model class represented by the five reactions ({1} to {5}). Gillespy2 [56] library was used to simulate 1000 trajectories of the cells. Figure 2.6 presents the trajectories of x_1 and x_2 for five out of the 1000 trajectories simulated, which represent five individual cells.

2.2.4 Simulation with chemical Langevin equation numerical approximation

As explained in Section 1.6.2, it has been demonstrated that the CME can be approximated by the CLE, described on Eq. (1.8). Therefore, the model class described by the CLE as an SDE system of:

$$\begin{aligned} x_1(t+dt) - x_1(t) =& (\alpha_1 - \beta_1 x_1)dt + (\sqrt{\alpha_1} N_1(t))\sqrt{dt} \\ & - (\sqrt{\beta_1 x_1} N_2(t))\sqrt{dt} \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} x_2(t+dt) - x_2(t) =& (\alpha_2 + \alpha_3 x_1 - \beta_2 x_2)dt + (\sqrt{\alpha_2} N_3(t))\sqrt{dt} \\ & + (\sqrt{\alpha_3 x_1} N_4(t))\sqrt{dt} - (\sqrt{\beta_2 x_2} N_5(t))\sqrt{dt}, \end{aligned} \tag{2.6}$$

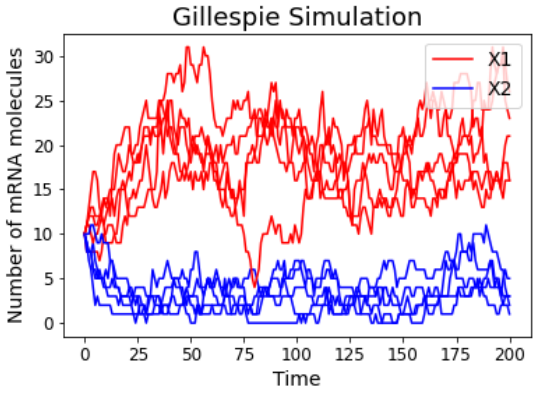


Figure 2.6: Gillespie's algorithm simulation of model class. Five trajectories of the numerical simulation for the evolution of the chemical reaction system (reactions {1} to {5)} by Gillespie's algorithm with the parameters from Table 2.1. Personally generated with differentSimulations.py script.

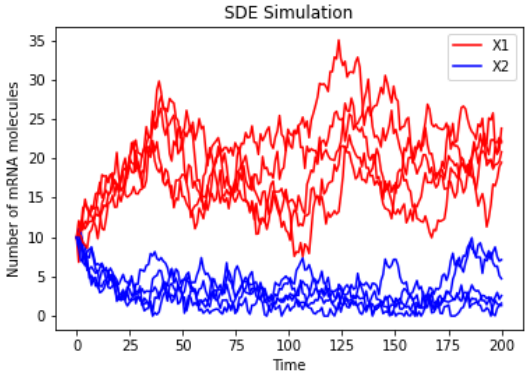


Figure 2.7: SDE numerical simulation of model class. Five trajectories of the simulation by numerically integrating the SDE system described by Eqs. (2.5) and (2.6) with the parameters from Table 2.1 and integration step of 1. Personally generated with differentSimulations.py script.

where the first part of the equation is the same as in the ODE system. The second part of the equation accounts for the stochasticity of the biochemical system, represented by N_i as a Gaussian noise for each reaction. This Gaussian noise, with mean 0 and standard deviation 1, is multiplied by the square root of the propensity of the reaction and by the square root of the integration step. Because the CLE is an SDE system, it was simulated by numerically integrating the system with a personal implementation of the Euler-Maruyama approximation.

The system was simulated 1000 times, to obtain 1000 trajectories of x_1 and x_2 mimicking a single-cell transcriptomic experiment with 1000 cells. The first five trajectories are presented in Figure 2.7.

To simulate trajectories of CLE with the Euler-Maruyama algorithm, two parameters are required: the number of simulations to perform and the integration time used. In order to develop a method based on the CLE, these two parameters have to be defined. On the one hand, these parameters affect the computational time of the algorithm, therefore it is preferred to have a lower number of simulations and higher integration step, to reduce the computational time of the simulation. On the other hand, a low number of simulations or a higher integration step could lead to undesired errors, thus, a balance needs to be achieved.

For this reason, the influence of the integration step (dt) and the number of cells simulated (N) was analyzed, and is presented in Figures 2.9 and 2.10. Appendix C presents a table with the results of numerical simulations used in the plots of these figures. Figure 2.8 presents the mean

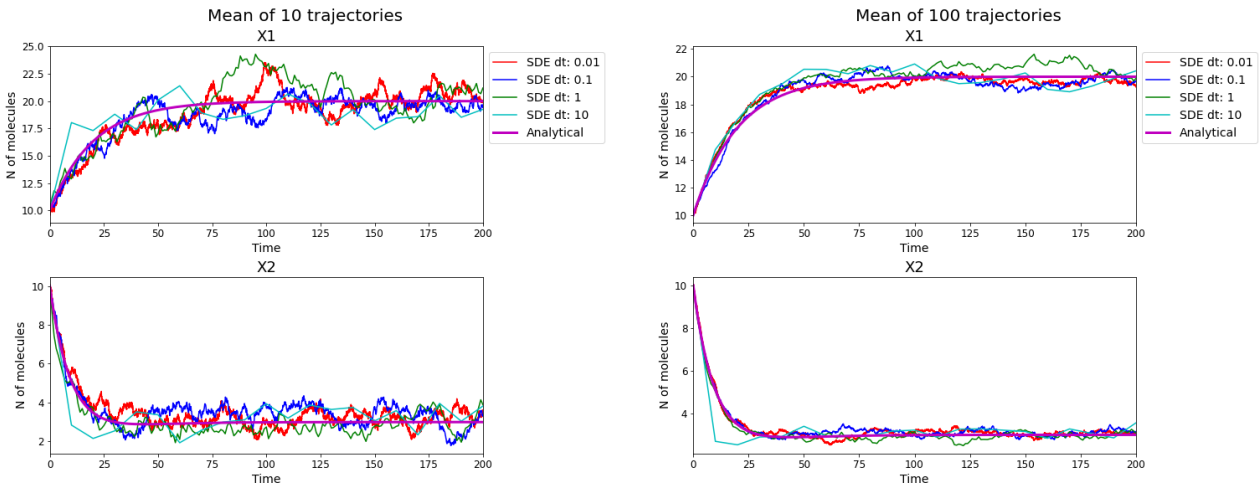


Figure 2.8: Mean trajectories of the numerically integrated SDE system for different number of trajectories simulated. Mean trajectories of 10 and 100 trajectories with different integration steps, when numerically integrating by Euler-Maruyama the SDE system of Eqs. (2.5) and (2.6) with initial conditions and parameters from Table 2.1. Mean trajectories for different number of cells and integration steps are available in Appendix D. Personally generated with evalODESimulations.py script.

trajectory for different numbers of cells simulated and different integration steps.

Analysis of integration step size

The SDE system was simulated with the Euler-Maruyama algorithm for different integration steps (0.01, 0.1, 1.0, 10.0). The computational time to run the simulation was recorded and the mean trajectory of the population of cells was compared with the analytical solution obtained in Section 2.2.1 by using:

$$\%Error = \frac{|x_{\text{Analytical solution}} - \overline{x_{\text{SDE simulation}}}|}{x_{\text{Analytical solution}}} 100. \tag{2.7}$$

The results are presented in Figures 2.9 and 2.10. In Figure 2.9, it can be observed that a smaller integration step makes the computing time of the simulation significantly longer. Also, from Figure 2.10, it can be said that an integration step of 10.0 has a higher error when compared with the analytical solution, but the other integration steps evaluated have similar errors. Therefore, an integration step of 1.0 was used for the rest of the project.

Analysis of number of cells to simulate

The SDE system was simulated for different number of trajectories (1, 5, 10, 50, 100, 500, 1000 and 5000). The time to run the simulation was recorded and the mean of the trajectory of the cells was compared with the analytical solution obtained in Section 2.2.1 by using Eq. (2.7). The results

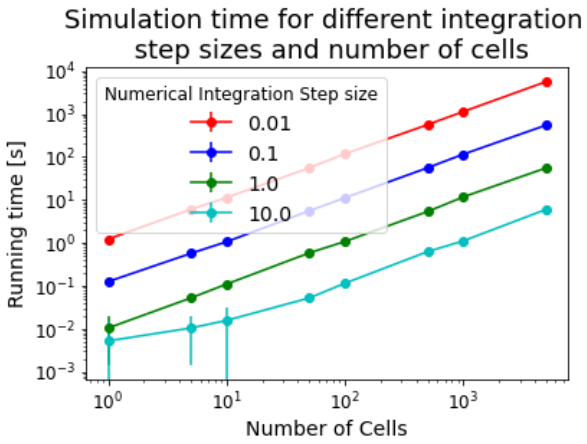


Figure 2.9: Computational time comparison to run the Euler-Maruyama scheme for numerical approximation of the SDE system. Comparison of the running time for different integration steps and different numbers of cells simulated when numerically integrating by Euler-Maruyama the SDE system of Eq. (2.5) and (2.6). Personally generated with evalODESimulations.py script.

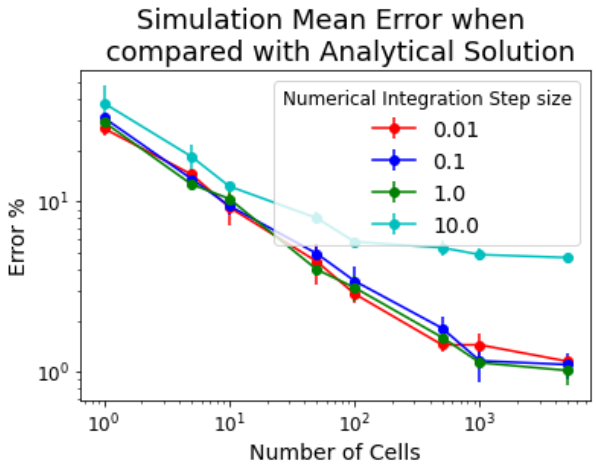


Figure 2.10: Simulation error of comparing Euler-Maruyama scheme for numerical approximation of the SDE against the analytical solution of ODE system. Comparison of the mean error for different integration steps and different numbers of cells simulated when numerically integrating by Euler-Maruyama the SDE system of Eq. (2.5) and Eq. (2.6). Personally generated with evalODESimulations.py script.

are presented in Figures 2.9 and 2.10. In Figure 2.9, it can be observed that a higher number of cells simulated makes the computing time of the simulation significantly longer. Also, from Figure 2.10, it can be observed that from 1000 cells, increasing the number of simulated cells does not decrease the percentage error. Therefore, the parameter 1000 simulations was used for the rest of the project.

2.2.5 Simulation methods comparison

The simulation methods CME, Gillespie's algorithm and CLE were compared by extracting snapshots at different time points (2, 5 and 10) and superposing their histograms (Figures 2.11 and 2.12). In these two figures, it can be observed that an ensemble of stochastic simulations of the Gillespie's algorithm and the CLE are good representations of the system when compared with the original probability distribution obtained by the CME, without the need of solving the 1600 ODEs system of the CME. Because the computing time of numerically integrating the CLE is less than the one for the Gillespie's algorithm, the SDE system will be used for the optimization algorithm to infer the reaction parameters. However, in the simulation of synthetic data for testing the performance

of the optimization and GRN inference methods that will be developed, computational efficiency is not an issue, and the Gillespie algorithm generates more realistic integer-valued data. Therefore, the Gillespie's algorithm will be used for this task.

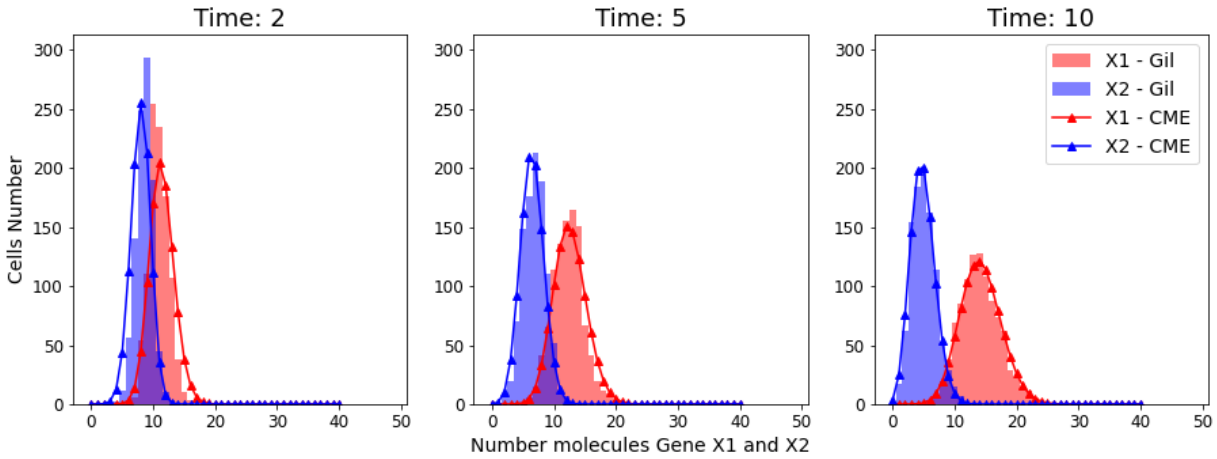


Figure 2.11: Gillespie's algorithm compared with the CME. Histogram of the trajectories snapshots at three different times (2, 5, and 10) of the Gillespie's simulation for 1000 trajectories (Section 2.2.3) and probability distribution by the CME (Section 2.2.2) at the same three times. Personally generated with differentSimulations.py script.

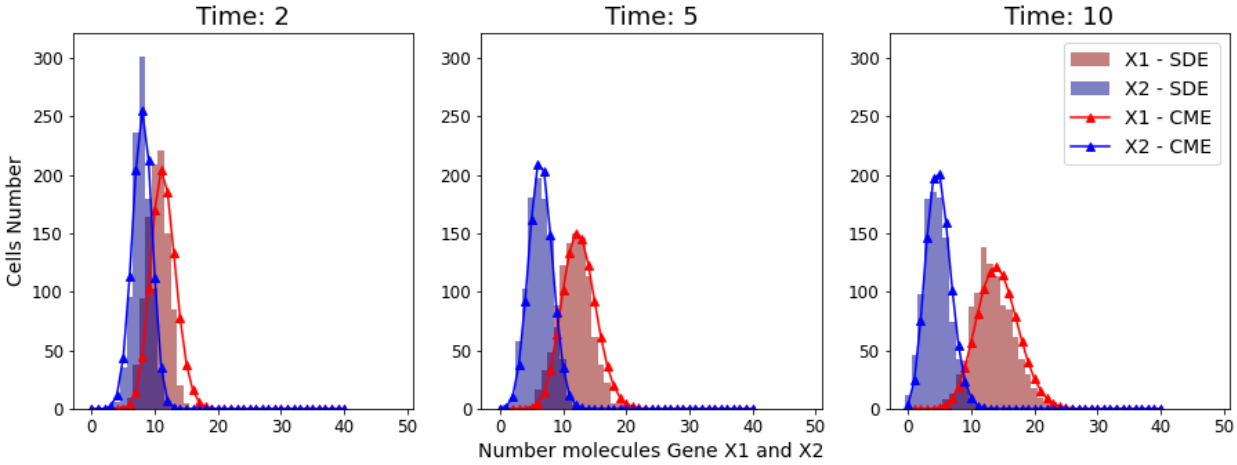


Figure 2.12: SDE numerical integration compared with the CME Histogram of the trajectories snapshots at three different times (2, 5, and 10) of the SDE (representing the CLE) numerical integration for 1000 trajectories (Section 2.2.4) and probability distribution by the CME (Section 2.2.2) at the same three times. Personally generated with differentSimulations.py script.

2.3 Generation of synthetic transcriptomics data

Chapters 3 and 4 of the manuscript use synthetically generated single-cell transcriptomics times series data to evaluate the performance of the developed methods. In these chapters, synthetically generated data is preferred over real data because the ground truth from which the data was generated is known, and can therefore be used to assess accuracy of the methods.

2.3.1 Synthetic data used in stochastic differential equation parameter estimation

Chapter 3 of the manuscript describes the optimization algorithm used to infer the parameters of a CLE system that makes the ensemble of trajectories simulated by Euler-Maruyama to approximate the distribution obtained in the single-cell transcriptomics time series data. To obtain the synthetic heterogeneous cell population expression data, the Gillespie algorithm for the same parameters and initial conditions as described in Table 2.1 was simulated for 5000 trajectories. Then, snapshots of these trajectories at five times (1, 5, 10, 20, 40) were extracted by sub-sampling 1000 of the 5000 cells at each of the five times, and the number of molecules of each gene simulated for each cell was stored. Figure 2.13 presents the histograms of the different snapshots for the RG (x_1) and the TG (x_2).

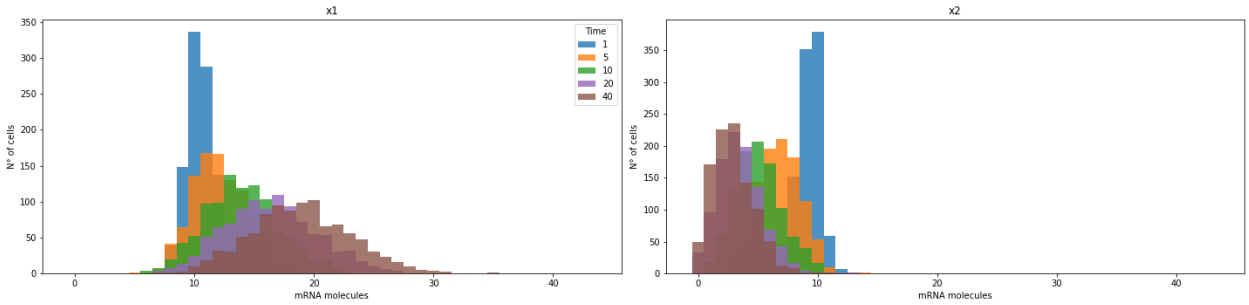


Figure 2.13: Snapshot of 1000 trajectories at 5 different times. Simulation of trajectories by Gillespie's algorithm used for SDE parameter estimation, with parameter and initial conditions defined in Table 2.1. Personally generated with 2_SDEoptimizationAlgorithm_5p.py script.

2.3.2 Synthetic data used evaluation of the developed method

Chapter 4 of the manuscript evaluates the performance of the GRN inference algorithm we developed when used to infer networks with different number of genes and topologies (Table 2.2). The synthetic data sets based on these networks were generated by using the Gillespie's algorithm for 5000 trajectories. Then, snapshots of these trajectories at five times (1, 5, 10, 20, 40) were

extracted by sub-sampling 1000 of the 5000 cells. The graphical representation of these GRNs, the mean and standard deviation of 5000 trajectories simulated by Gillespie's algorithm, and the histograms of the snapshots of the five times sampled are detailed in Appendix E.

Table 2.2: Networks simulated.

Network	N Genes	N Links
Network02_01	2	1
Network02_02	2	1
Network02_03	2	2
Network02_04	2	0
Network02_05	2	1
Network05_01	5	3
Network05_02	5	4
Network05_03	5	4
Network05_04	5	5
Network05_05	5	3
Network05_06	5	3
Network10_01	10	9
Network10_02	10	12
Network10_03	10	13
Network10_04	10	15

Chapter 3

Parameter estimation & scATA algorithm method development

With the objective of inferring causal regulatory relationships between the genes in the system, we have built an All-to-All (ATA) algorithm that evaluates all possible pairs of combinations of RG-TG. This algorithm is based on analyzing each pair of possible combinations by estimating the parameters of a SDE such that, by simulating many trajectories, the ensemble of those trajectories can reproduce the single-cell gene expression data obtained experimentally. The CLE was chosen over the CME and the Gillespie's algorithm because of its lower computational time, as described in Section 2.2.5. The first section of this chapter will focus on SDE parameter estimation, and the second section of the chapter will focus on the scATA algorithm.

3.1 Stochastic differential equation parameter estimation

Eqs. (2.5) and (2.6), representing the CLE, are used to infer the regulatory relationship between the RG (x_1) and the TG (x_2). To obtain the best possible ensemble of simulations that mimics the behaviour of the data, the five parameters of Eqs. (2.5) and (2.6) (α_1, β_1, α_2, α_3 and β_2) will be estimated. These parameters are estimated by fitting an ensemble of SDE simulations to the synthetic time series data simulated for heterogeneous cell population expression (Figure 2.13). The parameters are estimated in such a way to minimize the differences between the cell population distribution simulated by SDEs and the one from the synthetic data (mimicking real data).

3.1.1 Optimization Problem

The objective function (OF) used to measure the difference between the simulated distribution of cells and the one from the synthetic data is the square root of the sum of the square Wasserstein distance (WD) in each of the time points for both of the genes:

$$\begin{aligned} \min_{\alpha_1,\beta_1,\alpha_2,\alpha_3,\beta_2} \quad & \sqrt{\sum_{t\in T}\sum_{x=1,2} WD(Data_{x,t}, Estimated_{x,t})^2} \\ \text{s.t.} \quad & \alpha_1,\beta_1,\alpha_2,\alpha_3,\beta_2 \geq 0. \end{aligned} \tag{3.1}$$

In Eq. (3.1), **WD** is the Wasserstein distance in one dimension, **Data** is the distribution of the gene expression data for gene 1 and gene 2 and **Estimated** is distribution calculated by simulating 1000 trajectories (mimicking 1000 single-cells) of the SDE system with the estimated parameters and extracting snapshots of the data at the same time points as the real data. The optimization problem aims to find the model parameters ($\alpha_1, \beta_1, \alpha_2, \alpha_3, \beta_2$) that minimize the OF based on the WD because this minimum corresponds to the simulated distribution of cells being as close as possible to the synthetic data.

Wasserstein Distance

The Wasserstein distance, as it is a metric to calculate differences between distributions [55], is used to build the OF. In the OF, the square WDs is added for each of the time points for both of the genes, and then the root square is calculated. Figure 3.1 shows the distribution of number of molecules per gene for synthetic and estimated data. Additionally, on each histogram, the value of the WD is detailed. It can be observed that when the synthetic and the estimated data are generated with the same parameters (Table 2.1), the WD is smaller when compared to a different set of parameters ($\alpha_1, \beta_1, \alpha_2, \alpha_3, \beta_2 = 1, 1, 1, 1, 1$).

When trying to solve the original optimization problem to find the set of parameters ($\alpha_1, \beta_1, \alpha_2, \alpha_3$ and β_2) that minimize the OF described in Eq. (3.1), we faced a few challenges. In practice, the numerical optimization algorithm did not initially work. Therefore, we implemented two solutions already described in the literature that made the problem numerically easier to solve. These two solutions, scaling and noise initialization, will be described below.

Scaling

The plots in Figure 3.2 reflect how the value of the OF changes when one of the parameters is not the optimal one (i.e the one minimizing the OF). For each plot in this figure, all the parameters

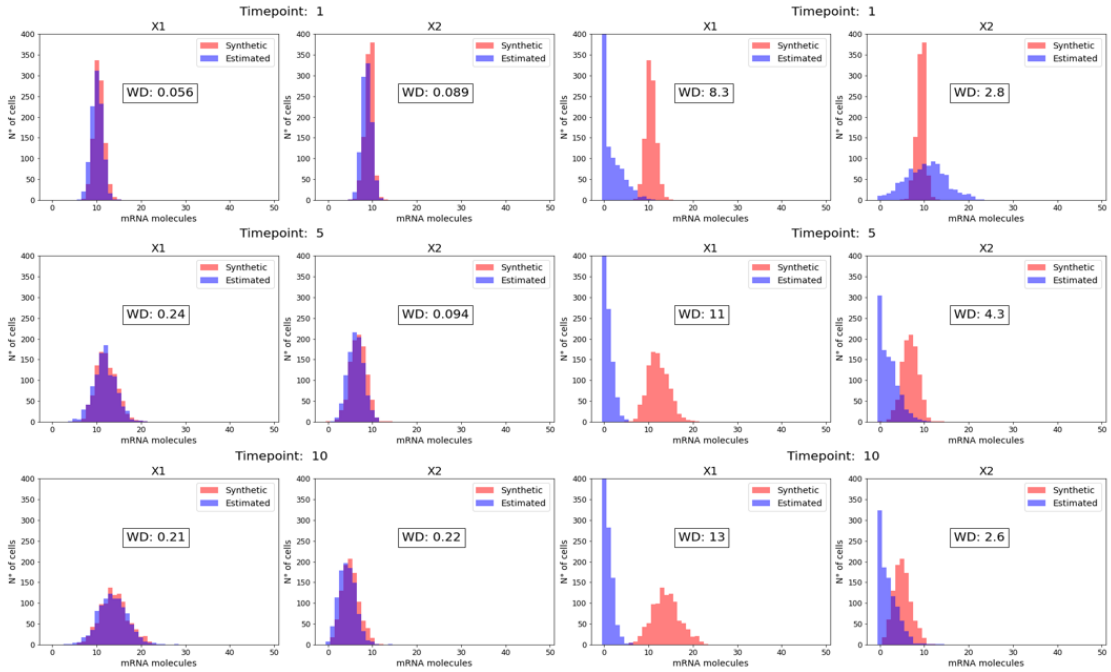


Figure 3.1: Distributions of synthetic data and snapshots of estimated data. Distributions of synthetic and estimated data for three time points analyzed. Synthetic data are the same as presented in Figure 2.13. Estimated data are snapshots of 1000 trajectories at each time point for the parameters defined in Table 2.1 for the left, and $\alpha_1, \beta_1, \alpha_2, \alpha_3, \beta_2 = 1, 1, 1, 1, 1$ on the left. The number inside each box is the WD between the distribution of the synthetic and the estimated data. Personally generated with 2_SDEoptimizationAlgorithm_5p.py

stayed as the ones used for the simulation of the synthetic data (Table 2.1) except for one that was changed, reflected across the x-axis.

As it can be observed in the first column of Figure 3.2, the scale of the parameters is different. For the SDE system used, β_1, α_3 and β_2 are multiplied by a variable, while α_1 and α_2 are not. This causes problems for the numerical optimization algorithm used to solve the problem. To solve this issue, inspired by [57], β_1, α_3 and β_2 were scaled by the average of the initial value of x_1 and x_2 of the data, depending on which of the variables it was multiplying. Therefore, the OF after the scaling is:

$$\min_{\alpha_1,\beta_1*\overline{x_1(0)},\alpha_2,\alpha_3*\overline{x_1(0)},\beta_2*\overline{x_2(0)}} \quad \sqrt{\sum_{t\in T}\sum_{x=1,2} WD(Data_{x,t}, Estimated_{x,t})^2} \qquad (3.2)$$

$$\text{s.t.} \quad \alpha_1, \beta_1, \alpha_2, \alpha_3, \beta_2 \geq 0.$$

The second column of Figure 3.2 shows how the scaled OF (Eq. (3.2)) changes when one of the parameters is altered. It can be observed in this second column that the five parameters which the optimization function is evaluating have similar numerical scales (i.e same order of magnitude),

allowing for numerical optimization techniques to perform better. Also, it can be observed that the parameter ranges in which the OF did not change much disappeared.

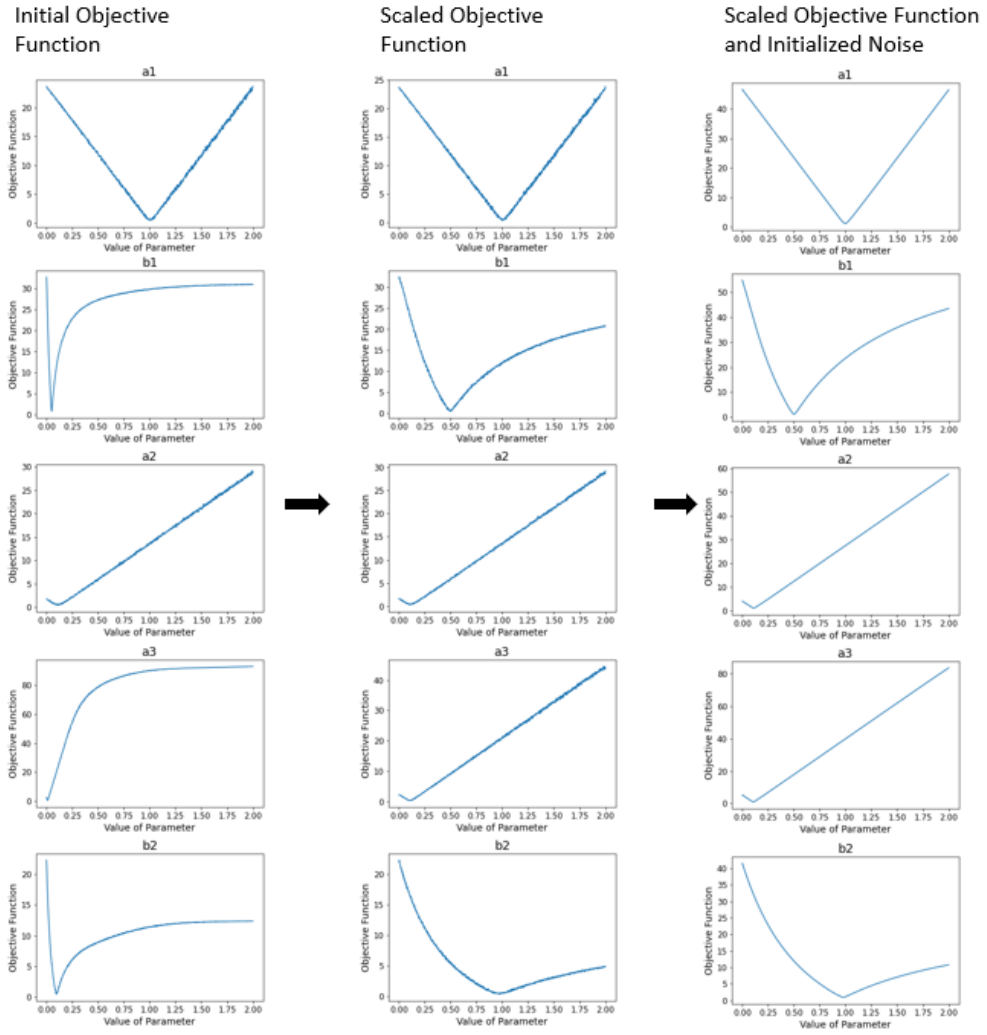


Figure 3.2: Objective functions evaluated. Plots of the OF when changing only one parameter. The remaining parameters stay with the value defined in Table 2.1. The first column is the OF without scaling or noise initialization, the second column is the OF scaled, and the third column is the OF scaled and the simulation running with initialized noise. This shows how the subsequent improvements of our approach, namely scaling and noise initialisation, changed the objective function making the optimization problem of finding its minima less difficult to tackle. The OF plotted for all different changes and for different ranges of the parameters is displayed in Appendix F. Personally generated with 2_SDEoptimizationAlgorithm_5p.py

Noise Initialization

The first and second columns of Figure 3.2 show fluctuations in the value of the OF, making it not being convex. These jumps are due to the noise in the SDE functions, where for every iteration

a sample from a Gaussian distribution with mean 0 and standard deviation 1 is taken for every reaction, cell and integration step.

To get rid of these local minima and transform the OF into a deterministic one, we used an approach previously described in [58], where the noise matrix is initialized beforehand. For this problem, we initialized the noise matrix as:

$$\text{Noise Matrix} = \begin{bmatrix} N_{1,1,1} & \cdots & N_{1,T,1} \\ N_{2,1,1} & \cdots & N_{2,T,1} \\ N_{3,1,1} & \cdots & N_{3,T,1} \\ N_{4,1,1} & \cdots & N_{4,T,1} \\ N_{5,1,1} & \cdots & N_{5,T,1} \end{bmatrix} \begin{bmatrix} N_{1,1,2} & \cdots & N_{1,T,2} \\ 2 & \cdots & N_{2,T,2} \\ 2 & \cdots & N_{3,T,2} \\ 2 & \cdots & N_{4,T,2} \\ 2 & \cdots & N_{5,T,2} \end{bmatrix} \begin{bmatrix} N_{1,1,C} & \cdots & N_{1,T,C} \\ C & \cdots & N_{2,T,C} \\ C & \cdots & N_{3,T,C} \\ C & \cdots & N_{4,T,C} \\ C & \cdots & N_{5,T,C} \end{bmatrix}, \tag{3.3}$$

and then all the trajectories will be simulated with the same initialized noise matrix. The improvement on the shape of the OF can be observed in the last column of Figure 3.2, where now the functions are smoother and do not present the multitude of local minima which were observable in the other columns. The effects of scaling and noise initialization can be observed in the figures from Appendix F, where the same OF evaluations from Figure 3.2 are performed for different ranges of the parameters. Especially when the range is smaller, the effect of the noise initialization is clearly evident.

3.1.2 Optimization Algorithm

The optimization problem is solved by using the scipy.minimize python library with the L-BFGS-B method, a quasi-Newton algorithm [59] for different boundaries and initial conditions of the parameters. The different OFs (scaled and not scaled) and the two simulation routines (with and without initialized noise) were compared. Figure 3.3 presents the optimization results for the different optimization problems evaluated for different initial conditions and boundaries of the parameters. In this figure, it can be observed that the best combination (i.e. the combination that leads most of the times to a smaller OF value) is the scaled OF with an initialized noise simulation algorithm.

Also, it can be observed that even with a scaled-OF and initialized noise, for some combinations of initial conditions and boundaries of the parameters, the optimization algorithm is not able to reach the minimum value. This means that it is still getting trapped in local minima. To solve this issue, global optimization algorithms were evaluated, but the computational time taken to run them was too

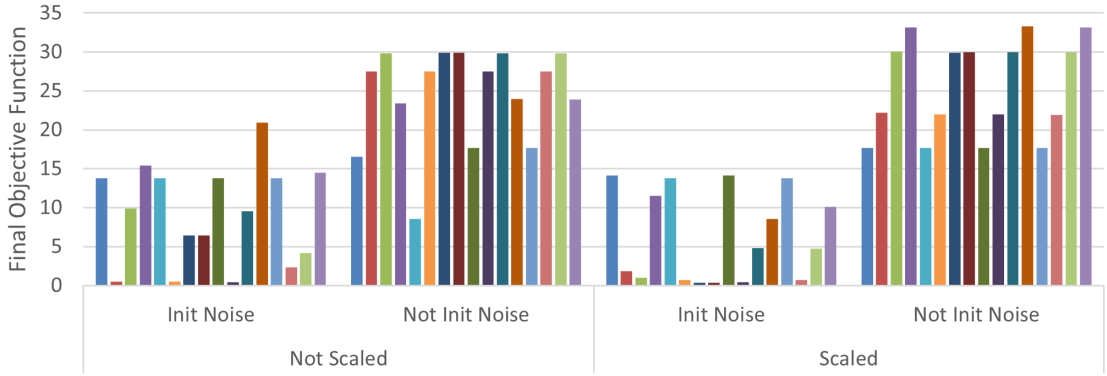


Figure 3.3: Comparison of OFs to get to a minimum value. Value obtained when minimizing the different OFs with the different simulation schemes, from different initial conditions of the parameters and different boundaries. Each color represents a different initial condition and boundary combination. Figure made in excel with personally generated data with 2_SDEoptimizationAlgorithm_5p.py

long to be practically useful in developing our method, which we want to be scalable to evaluate a very large number of potential gene regulations. Thus, we decided instead to employ several parameter initial conditions on the final scATA algorithm.

3.2 Single-cell All-to-All (scATA) algorithm

The objective of the scATA algorithm which we developed in this thesis and which we describe in Section 3.2.6 is to infer a GRN from single-cell transcriptomic data by analyzing interactions between all pairs of genes. The aim is to infer the regulatory relationship between each couple of RG and TG. Therefore, it is not necessary to study the whole SDE system described with Eqs. (2.5) and (2.6), but just the equation which contains the regulation.

3.2.1 Objective function and stochastic differential equation

Because of the interest only on the regulatory relation from the RG to the TG, the OF of the ATA algorithm will only consider the WD between the simulated distribution and the distribution from the data for x_2:

$$\min_{\alpha_2,\alpha_3*\overline{x_1(0)},\beta_2*\overline{x_2(0)}} \quad \sqrt{\sum_{t\in T} WD(Data_{2,t}, Estimated_{2,t})^2} \qquad (3.4)$$
$$\text{s.t.} \quad \alpha_2, \beta_2 \geq 0.$$

As we also intent to identify negative regulation, the parameter α_3 can also be negative. The Euler-Maruyama algorithm does not allow parameters to be negative, therefore, we will generate

the simulated data with two equations, depending if α_3 is negative or not. If α_3 is positive, the SDE used to simulate the trajectories and extract the distribution of x_2 at the different time points for the final scATA algorithm will be:

$$\begin{aligned} x_2(t+dt) - x_2(t) =& (\alpha_2 + \alpha_3 * x_1 - \beta_2 * x_2)dt + (\sqrt{\alpha_2} * N_3(t)) * \sqrt{dt} \\ &+ (\sqrt{\alpha_3 * x_1} * N_4(t)) * \sqrt{dt} - (\sqrt{\beta_2 * x_2} * N_5(t)) * \sqrt{dt}. \end{aligned} \tag{3.5}$$

On the other hand, if α_3 is negative, we will use:

$$\begin{aligned} x_2(t+dt) - x_2(t) =& (\alpha_2 + \alpha_3 * x_1 - \beta_2 * x_2)dt + (\sqrt{\alpha_2} * N_3(t)) * \sqrt{dt} \\ &- (\sqrt{|\alpha_3| * x_1} * N_4(t)) * \sqrt{dt} - (\sqrt{\beta_2 * x_2} * N_5(t)) * \sqrt{dt}, \end{aligned} \tag{3.6}$$

where the parameter of α_3 is now considered to be contributing to the degradation of x_2,

The aim of the optimization method is to obtain the best fit of the ensemble of simulations for the TG (x_2) to the data by estimating only parameters α_2, α_3 and β_2. Then, for each OF evaluation of the algorithm, the trajectories of the cells will be simulated by Eq. (3.5) or Eq. (3.6). By reducing the number of free parameters, we have now simplified the optimization problem without loosing anything in terms of applicability to our biological problem. However we have considerably gained in terms of a less complex optimization problem to solve, and a faster time for the optimization to converge, important factors for our final method to be up scalable to infer many gene interactions.

3.2.2 Regulator Gene Interpolation

Eqs. (3.5) and (3.6) simulate the number of molecules of the TG for each cell at any time point based on the three parameters and the gene expression of the RG. Because the integration step of the Euler-Maruyama implementation for the numerical simulation of the SDE over time is much smaller than the separation between snapshot time-points on the data, the expression of the RG (x_1) needs to be obtained for each of the integration steps. With this purpose, the expression of x_1 will be interpolated over all the time points.

To perform this interpolation, the cells at each time point will be sorted by the number of molecules of x_1 from the cell with the highest number of molecules to the cell with the smallest. Then, the cell with the highest number of mRNA molecules of the RG on the first time point will be connected with the one with the highest number of RG mRNA molecules at the following time points. Subsequently, the cell with the second highest number of mRNA molecules of the RG on the first time point will be connected with the one with the second highest number of RG mRNA molecules at the following time points, and so on so forth for all the sorted cells (Figure 3.4 left). Finally, the connected cells are interpolated with a cubic interpolation (Figure 3.4 right), and the gene expression of the cell for

each integration time step is obtained. At the moment, this interpolation is only implemented for samples of time points with the same number of cells. This interpolation is only performed over the RG because its expression does not depend on the expression of the TG.

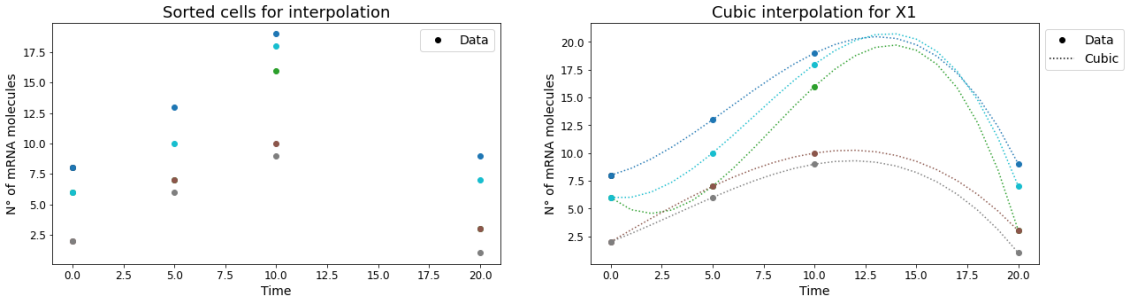


Figure 3.4: Interpolation of x_1 Left: The cells are sorted by number of mRNA molecules for the RG and connected. Right: The connected cells are interpolated and the expression of the gene for each integration step is obtained. Personally generated with gene_interpolation.py

3.2.3 Initial condition for target gene trajectories

The information of the first time point will be used as initial number of molecules for the TG to simulate the trajectories. For the sorted cells (for x_1) of the first time point, the value of x_2 will also be stored. Each x_2 value will be used as $x_2(0)$ to start the numerical simulation with Eqs. (3.5) or (3.6) and the $x_1(t)$ used to propagate that trajectory will be the one that matches that cell.

3.2.4 Evaluate the regulatory improvement

To test if the directed regulatory relation from the RG to the TG exists, two scenarios will be compared: with and without regulation. For this, two different optimization problems will be solved. In the first one, it will be assumed that there is no regulation (i.e. the value of α_3 on Eq. (3.5) is 0) and the OF of the best fit of the ensemble of simulations will be saved. This optimization algorithm will start from different initial conditions of α_2 and β_2 ([0, 0], [0.1, 0.1] and [1, 1]). Then, on the second optimization problem, the restriction of α_3 being 0 will be lifted, and the optimization program will run again from different initial conditions of α_2, α_3 and β_2 ([0, 0, 0], [1, 0, 1], [0.1, 0.1, 0.1] and [1, 1, 1]). The minimum value of both optimization problems from different initial conditions will be stored and compared in different ways. These different comparison methods are detailed on Section 3.2.5.

3.2.5 Evaluated scores

The aim of the ATA approach is to try estimating the three parameters (α_2, α_3 and β_2) for all pair of RG-TG combinations. Therefore, after this has been done for every pair of genes, in order to infer the GRN, we have to determine which of these models are saying there is a regulatory link and for which pair of genes. To do this, we built a scoring system with different possible ways to say if a regulatory link exists. All of the scores presented below represent different ways to determine which of the links, from out of all the possible RG-TG combinations can be extracted from the data, and were evaluated as potential score candidates.

Regulation objective function (OF)

Optimal value of the OF when solving the optimization problem with regulation (α_3 not being restricted to 0). A lower OF score is preferred over a higher OF score.

Regulation Parameter α_3

Absolute value of the regulatory parameter when solving the optimization problem with regulation (α_3 not being restricted to 0). A higher α_3 score is preferred over a lower α_3 score.

Difference

Difference between optimal OF with (α_3 not being restricted to 0) and without regulation (α_3 restricted to 0), calculated as:

$$\text{Difference} = OF_{\text{regulated}} - OF_{\text{notRegulated}}. \tag{3.7}$$

A lower score is preferred over a higher score.

Division

Quotient of the optimal OF with regulation (α_3 not being restricted to 0) divided by the optimal OF without regulation (α_3 restricted to 0), calculated as:

$$\text{Division} = \frac{OF_{\text{regulated}}}{OF_{\text{notRegulated}}}. \tag{3.8}$$

A lower score is preferred over a higher score.

Improvement

Improvement of the OF when regulation is considered (α_3 not being restricted to 0) compared to when regulation is not considered (α_3 restricted to 0), calculated as:

$$\text{Improvement} = \frac{OF_{\text{regulated}} - OF_{\text{notRegulated}}}{OF_{\text{notRegulated}}}. \tag{3.9}$$

A lower score is preferred over a higher score.

Objective function (OF) and Regulation Parameter α_3

When the OF is low, it means that the proposed model is fitting well the data. In some cases, this low OF is achieved with an α_3 of 0, meaning that the model fits well the data, but there is no regulation. To avoid this cases when comparing the OF values, when α_3 is 0, the OF will be automatically set at 100 (a number very big compared with the rest of the values). A lower score is preferred over a higher score. By construction, this score is meant to perform better than the OF alone, and in fact it will outperform OF systematically in Chapter 4.

Improvement and Regulation Parameter α_3

For the same reason as before, this score sets the value of Improvement score to be 100 (a number very big compared with the rest of the values) if the value of α_3 in the optimization problem with regulation (α_3 not being restricted to 0) is 0. A lower score is preferred over a higher score.

Improvement, Rank and Regulation Parameter α_3

To consider the previous knowledge that genes are regulated by few other genes, the rank metric sorts by improvement and gene regulated the connections, and then ranks the regulators of each TG giving a higher score to the best regulator (better improvement).

The value of Improvement and $\alpha 3$ multiplied by rank to account for the fact that genes are regulated by few other genes but also consider the improvement, calculated as:

$$\text{Improvement + } \alpha_3\text{+ rank} = (\text{Improvement +}\alpha_3)\text{rank} \tag{3.10}$$

A lower score is preferred over a higher score.

A detailed analysis of the results given by these scores on synthetic data is presented in Chapter 4.

3.2.6 Developed scATA algorithm

The following pseudo-code summarizes all the steps described previously and describes the final implementation of the method developed to infer GRN from single-cell transcriptomic data, as presented in Figure 1.5. It is implemented on the python scripts named **3_scATA_noReg_parallel.py**, **3_scATA_Reg_parallel.py** and **4_evaluate_scATA.py**, delivered in the supplementary materials of this thesis. It is programmed to be run with parallel computing, and the number of cores has to be defined on both scripts: **3_scATA_noReg_parallel.py** and **3_scATA_Reg_parallel.py**.

Developed algorithm: scATA

Step 1: Define which genes to test as regulator and target genes.
It can be all genes in the transcriptomic data.

Step 2: **For each combination of regulator - target genes:**

2.1 Interpolate the expression of the regulator gene (Section 3.2.2)

2.2 Estimate the parameters α_2 and β_2 for $Eq.$ (3.5) of the target gene without regulation ($\alpha 3 = 0$).
Run SDE optimization algorithm (Section 3.1.2) from different initial conditions of the parameters fixing $\alpha 3$ to be 0 always.

2.3 Estimate the parameters α_2, α_3 and β_2 for $Eqs.$ (3.5) and (3.6) of the target gene with regulation.
Run SDE optimization algorithm (Section 3.1.2) from different initial conditions of the parameters.

2.4 Save the best result (minimum value of the objective function) from steps 2.2 and 2.3.

Step 3: Calculate the evaluation scores (Section 3.2.5).

Step 4: For scores *improvement* and *improvement*+α_3+*rank* select the highest scores to propose as candidate links.

Chapter 4

Evaluation of scATA method on synthetic data

The performance in inferring GRNs of the single-cell All-to-All (scATA) algorithm which we developed in this thesis was evaluated by applying it to several synthetically generated GRNs with different numbers of genes (2, 5 and 10) and different topologies. To mimic the experimental results obtained by a single-cell transcriptomic experimental technique, for each considered network, 5000 cells were simulated independently by using the Gillespie's algorithm as described in Section 2.3.2. Then, snapshots of the population were taken at five different time points (1, 5, 10, 20 and 40). The GRNs graphical representation, the mean and standard deviation of the trajectories of the genes expressions, and the histograms of the snapshots of these simulations are presented in Appendix E.

Our algorithm, detailed in Section 3.2.6 and implemented in the python scripts **3_scATA_noReg_parallel.py**, **3_scATA_Reg_parallel.py** and **4_evaluate_scATA.py**, was applied to the synthetic data sets with the different underlying GRNs. The purpose of this part of the study is to evaluate the performance and accuracy of scATA at inferring the pre-designed GRNs, where the ground truth is known, and the data contains no measurement noise. For each network to which we apply our method to, the output of the pipeline is a table of scores for each of the possible links, like the ones presented as an example in Appendix G. The results of the evaluation of the scATA algorithm on synthetic data are presented in this chapter.

4.1 Evaluation Metric

The area under the receiver operating characteristic curve (AUROC) will be used to evaluate the accuracy of the algorithm into predicting the true links of the synthetically generated data sets with known underlying GRNs. The AUROC is a standard tool to assess performance and has been used in similar studies, where GRN inference algorithms are evaluated [25]. The receiver operating characteristic curve is built by calculating the true positive rate (TPR):

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{4.1}$$

and the false positive rate (FPR):

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{4.2}$$

at every possible threshold of the score evaluated. The scores evaluated are defined on section 3.2.5.

The TPR, described in Eq. (4.1), is calculated by counting at every threshold of the score how many links are correctly said to exist (true positive (TP)) over how many links exist in the network, which is equivalent to the number of TP links plus the number of false negative (FN) links. The FPR is calculated in Eq. (4.2) by counting at every threshold of the score how many links are wrongly said to exist (false positive (FP)) over how many of the links evaluated do not exist in the network, which is equivalent to the number of FP links plus the number of true negative (TN) links. For example, in Figure 4.2, the pink line of the top panel on the left was built by counting how many TP, FN, FP and TN were for each *Improvement* score threshold.

The AUROC is, as its name says, the area under the ROC curve, and it is a way to summarize the overall curve into one number. An AUROC of 1 means it is a perfect reconstruction of the network and an AUROC of 0.5 means the method is performing the same as a method doing a completely random choice. Thus, an AUROC value between 0.5 and 1 is better than randomly guessing, but not perfect, and the higher the AUROC is, the better the algorithm. The python library sklearn was used for the drawing of the ROC curves and for calculating the AUROC values presented below.

4.2 2-genes networks

Five different 2-gene GRNs were evaluated with the aim of assessing if the SDE parameter estimation algorithm was working in an ATA format. The topologies, mean trajectories and time point

snapshots of these networks are presented in Appendix E. For three of the networks, the ROC curve of the *Improvement* score is presented in Figure 4.1 and the AUROCs for each of the scores evaluated are detailed in Table 4.1.

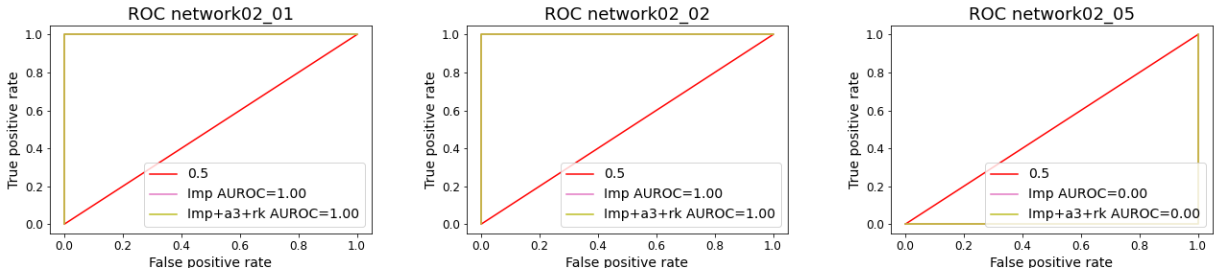


Figure 4.1: ROC curve for Improvement score for networks of 2 genes. The three figures present the ROC curve for the data set generated by GRNs of 2 genes. In every panel, the AUC of the ROC curve can be observed. Personally generated with 4_evaluate_scATA.py

Table 4.1: AUROC of all the scores evaluated for networks of 2 genes. OF Reg.: Value of optimal objective function when regulation is considered, α_3 Regulation parameter α_3 from Eqs. (3.5) and (3.6), Dif.: Difference, Div.: Division, Imp.: Improvement.

	AUROC							
Network	OF Reg.	α_3	OF & α_3	Dif.	Div.	Imp.	Imp. & α_3	Imp., α_3 & Rank
02_01	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00
02_02	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
02_03	-	-	-	-	-	-	-	-
02_04	-	-	-	-	-	-	-	-
02_05	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The first two networks (02_01 and 02_02) have the same topology (Appendix E), and just differ in the parameters used to perform the simulations, while the third network (02_05) was simulated by using a GRN with negative regulation (repression) from gene 1 to gene 2. In the first two networks, all the possible scores, excluding the *difference* score for network 02_01, were able to detect the true link before the other link, while in the third network, it did not. The ROC curves of all the different scores considered are presented in Appendix H, but, as it might be observed from the tables of this chapter, *Improvement* and *Improvement with* α_3 *and ranking* were performing better, thus they are presented in the figures. There are two networks, 02_03 and 02_04, that are not presented in the figure and have missing values (-) in the table. This is because they had all the possible links (02_03) or no links at all (02_04), and therefore the values of the FPR or TPR by definition cannot

be calculated. For these two networks, the complete scoring tables are presented in Appendix G. In those tables, it is possible to see that one link could be detected for network 02_03 and no links were detected for network 02_04.

4.3 5-genes networks

Figure 4.2 shows the ROC curves for the *Improvement* and *Improvement with* α_3 *and ranking* scores of the scATA algorithm applied to all synthetic data sets generated from GRN of five genes. The topologies, mean trajectories and time point snapshots of these networks are presented on Appendix E, and the ROC curves for the rest of the scores can be found in Appendix H. As it can be observed in Figure 4.2, for the first three and the last networks, the algorithm performed better than for the other two. As it can be observed in Appendix E, the topologies of networks 05_01, 05_02 and 05_03 did not include multiple-input systems or negative regulation, which might make the identification of the gene regulations more difficult, and thus may have caused the difference in the performance of the algorithm. Although, network 05_06 does include negative regulation, and the scATA algorithm had an AUROC of 0.82 for the *Improvement* score.

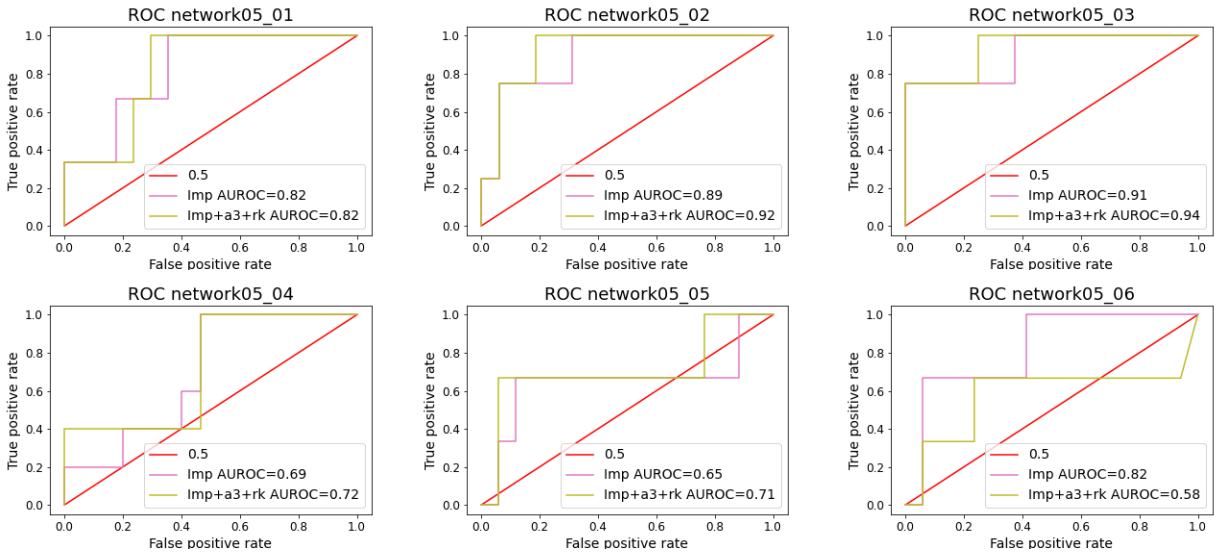


Figure 4.2: ROC curve for Improvement score for networks of 5 genes. The six figures present the ROC curve for the data set generated by GRNs of 5 genes. In every panel, the AUC of the ROC curve can be observed. Personally generated with 4_evaluate_scATA.py

Table 4.2 presents the AUROC values for all the different scores evaluated. In this table, it is possible to notice that score optimal value of the *OF* when regulation is considered (*OF Reg.*) does not perform as other scores, and that scores *Division* (Div.) and *Improvement* (Imp.) have the same

value. Finally, coupling the *Improvement* score with α_3 or the Rank does not always improve the classification.

Table 4.2: AUROC of all the scores evaluated for networks of 5 genes. OF Reg.: Value of optimal objective function when regulation is considered, α_3 Regulation parameter α_3 from Eqs. (3.5) and (3.6), Dif.: Difference, Div.: Division, Imp.: Improvement.

	AUROC							
Network	OF Reg.	α_3	OF & α_3	Dif.	Div.	Imp.	Imp. & α_3	Imp., α_3 & Rank
05_01	0.39	0.77	0.41	0.78	0.82	0.82	0.82	0.82
05_02	0.69	0.74	0.77	0.75	0.89	0.89	0.91	0.92
05_03	0.56	0.77	0.63	0.81	0.91	0.91	0.91	0.94
05_04	0.39	0.77	0.52	0.71	0.69	0.69	0.69	0.72
05_05	0.35	0.68	0.41	0.67	0.65	0.65	0.69	0.71
05_06	0.47	0.56	0.32	0.80	0.82	0.82	0.52	0.58

4.4 10-genes networks

Figure 4.3 presents the ROC curve for the scATA algorithm applied to the data sets simulated with GRNs of 10 genes. The topologies, mean trajectories and time point snapshots of these networks are presented in Appendix E. When compared with Figure 4.2, it can be seen that the scores are lower, meaning it is more difficult for the algorithm to find the true links when there are more genes and more links to try in an ATA methodology. While most of the *improvement* scores ranged from 0.60 to 0.71, the data set generated with GRN 10_02 had the lowest score of 0.54.

Table 4.3: AUROC of all the scores evaluated for networks of 10 genes. OF Reg.: Value of optimal objective function when regulation is considered, α_3: Regulation parameter α_3 from Eqs. (3.5) and (3.6), Dif.: Difference, Div.: Division, Imp.: Improvement.

	AUROC							
Network	OF Reg.	α_3	OF & α_3	Dif.	Div.	Imp.	Imp. & α_3	Imp., α_3 & Rank
10_01	0.52	0.62	0.60	0.59	0.64	0.64	0.65	0.66
10_02	0.50	0.52	0.55	0.51	0.54	0.54	0.57	0.60
10_03	0.60	0.56	0.56	0.55	0.60	0.60	0.59	0.59
10_04	0.39	0.69	0.58	0.70	0.71	0.71	0.74	0.76

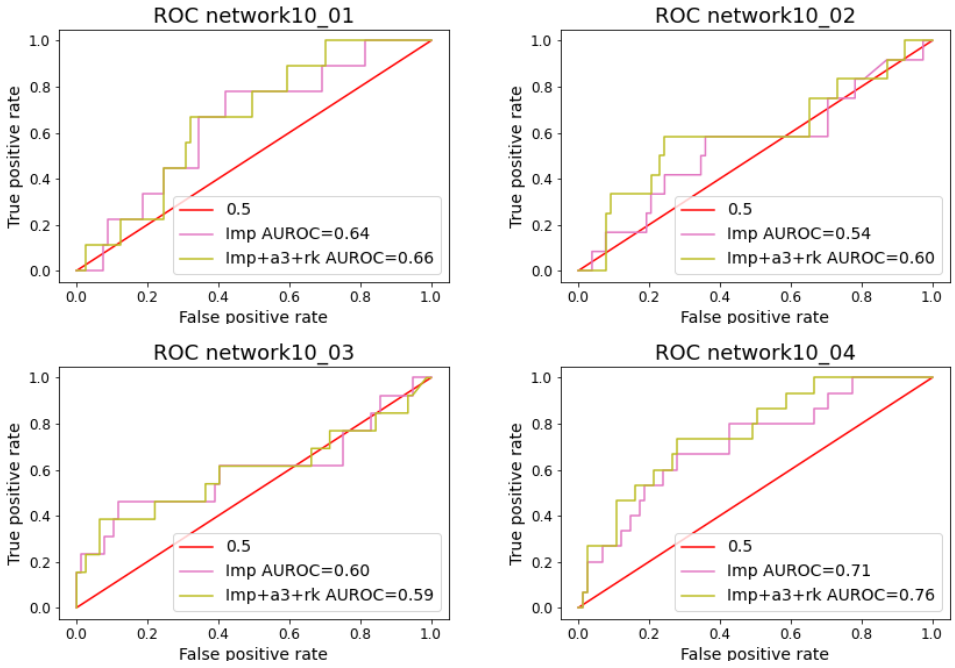


Figure 4.3: ROC curve for Improvement score for networks of 10 genes. The four figures present the ROC curve for the data set generated by GRNs of 10 genes. In every panel, the AUC of the ROC curve can be observed. Personally generated with 4_evaluate_scATA.py

Table 4.3 shows the same trend as Table 4.2, where the *OF Reg.* score has a lower performance than the other scores. This score is improved when coupled with the α_3 regulatory parameter, but is still lower than the others. It can be observed on Appendix H that these two scores are below the 0.5 line for all of the networks analyzed with 10 genes. On the other hand, the *Improvement* and *Improvement with* α_3 *and ranking* scores had AUROC values always over 0.5, and for most networks over 0.6.

It can also be observed from the panels in Figures 4.2 and 4.3, that the shape of the ROC curves has a high increase for lower thresholds. This means that for low threshold of the *Improvement* and *Improvement with* α_3 *and ranking* scores, our scATA method proposes regulatory links with a high true positive rates and a low false positive rates. Therefore, these two last scores were chosen to continue the further analysis of the application of scATA algorithm in a real data set (Chapter 5).

Chapter 5

Evaluation of scATA method on real data

The aim of this chapter is to apply our GRN inference algorithm to the single-cell transcriptomic data set obtained by [43]. The scATA algorithm will be applied in order to test its performance with a data set that has a known ground truth GRN and to illuminate the gene regulations occurring in the underlying biological system itself.

5.1 Erythroid-myeloid-lymphoid cell differentiation data set

During the study of stability and cell fate decision in the cell differentiation process, [43] analyzed the differentiation process of blood progenitors erythroid-myeloid-lymphoid (EML) cells into myeloid (MYL) cells and erythroid (ERY) cells. In this study, the cells were treated either with GM-CSF/IL-3 to induce the differentiation into MYL cells, with EPO to induce the differentiation into ERY cells, or with both GM-CSF/IL-3 and EPO. In the study, samples of the EML cells were analyzed before treatment (day 0), and cell samples from the three different treatments were obtained at days 1, 3 and 6, as it can be observed in Figure 5.1. For each of the samples, the single cells were sorted and the mRNA of 19 genes (17 from the GRN studied and 2 housekeeping genes) were measured by OpenArray qPCR. The obtained single-cell expression data were analyzed with OpenArray qPCR analysis software, and then, the quality of the data was analyzed. The detailed protocol of the study can be found in [43].

The time series data set analyzed in this thesis is the one published after pre-processing, and was composed of 10 samples of 150 or 200 single-cells per each time-point and treatment (Table 5.1).

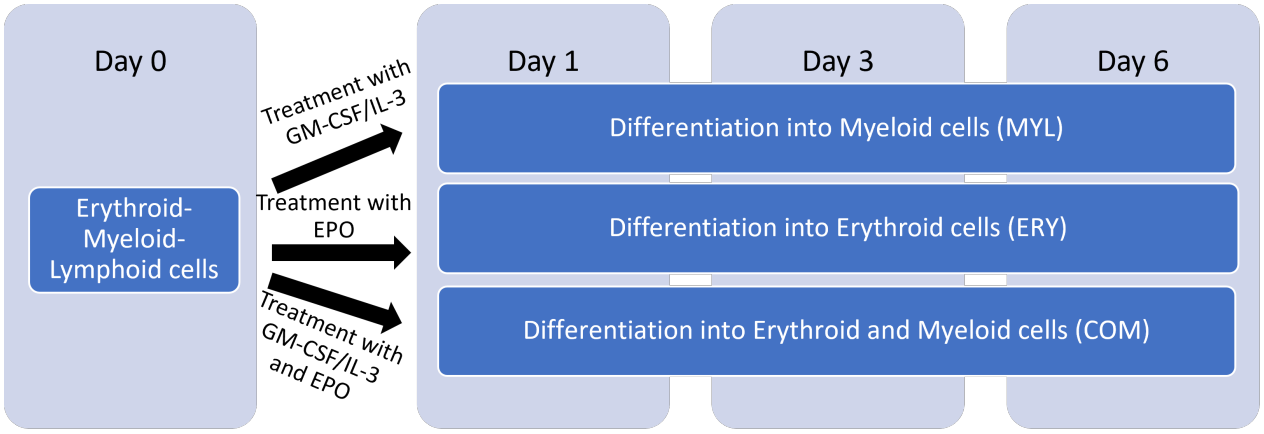


Figure 5.1: Experimental treatments to obtain samples. Cell differentiation experiment, single-cell RT-qPCR and data pre-processing done by [43].

Table 5.1: Samples and number of cells per sample analyzed. Data set obtained directly from [43]

		Number of cells			
Treatment	Name	Day 0	Day 1	Day 3	Day 6
No treatment	EML	150	0	0	0
GM-CSF/IL-3	MYL	0	150	200	150
EPO	ERY	0	150	200	150
GM-CSF/IL-3 & EPO	COM	0	150	150	150
All Cells	ALL	150	450	550	450

The data set presented in Table 5.1 was modified in order to fulfil the condition to have the same number of cells in each of the time points, because the scATA algorithm currently requires that. To do this, the number of cells at day 3 for cells MYL and ERY was reduced to 150 by sub-sampling 150 cells out of the 200 randomly. Additionally, the control cells (EML) were used as day 0 for the three different treatments, as in the original study. When analyzing all the cells, the data of day 0 was considered three times to obtain 450 cells. The result was a time series data set composed of 150 cells for each time point for samples MYL, ERY and COM, and a time series data set of 450 cells for each time point for data set ALL.

5.2 Preliminary data analysis

The gene expression of the different treatments and samples was analyzed, and the histograms per gene of each of the treatments, and from all the cells merged, are available in Appendix I. In

these histograms, it is possible to observe how for some samples, some genes reduce or increase the number of molecules of mRNA available.

For all cells in different time points (Appendix I), it is possible to observe a reduction in the mRNA molecules of Gata2, Runx, Fli1, Scl, cMyb, ckit and TBP from the untreated sample in comparison to the rest of the days. On the other hand, for all the cells but comparing different treatments (Appendix I), a difference between samples is observed on genes sfp1, Gata2 and EpoR. It can also be observed from the histograms that there are several genes which have a high number of cells for which zero molecules have been detected for them.

Principal component analysis (PCA) (a standard dimensionality reduction technique) was applied to the different data sets in order to visualize the distribution of the measured transcriptome of the cells, composed of the 19 genes, in a 2-dimensional space. The coordinates for the PCA plots presented in Figures 5.2 and 5.3 were determined with the python library sklean.

Figure 5.2 shows the PCA of all the cells analyzed in the sample, separated by time or treatment. In both of its panels, it is possible to see how the cells change their transcriptome over time, and separate from the control sample. The figure on the left shows how by day 6, there are three different clusters. Because the cells are located on the same position on both figures, it can be observed on the figure on the right that one of the clusters is composed of cells treated with GM-CSF/IL-3 alone and GM-CSF/IL-3 combined with EPO, while the other two clusters are composed mainly of cells treated with EPO, but also some cells treated with GM-CSF/IL-3.

Figure 5.3 shows the PCA visualization for the samples over time for the different treatments separately. For this figure, the PCA dimensionality reduction was performed again, so the positions of the cells between panels have no relation between each others or with the position of the cells in

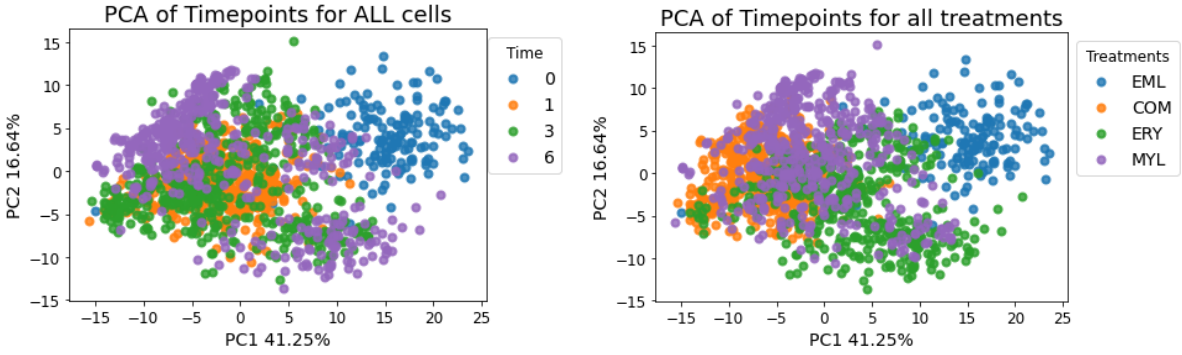


Figure 5.2: PCA projection of all the cells analyzed. Cell were separated by time point (left) and by treatment (right). ALL: all cells analyzed, COM: cells treated with GM-CSF/IL-3 and EPO, ERY: cells treated with EPO, MYL: cells treated with GM-CSF/IL-3. Personally generated with 1_scAnalysis_BloodData.py

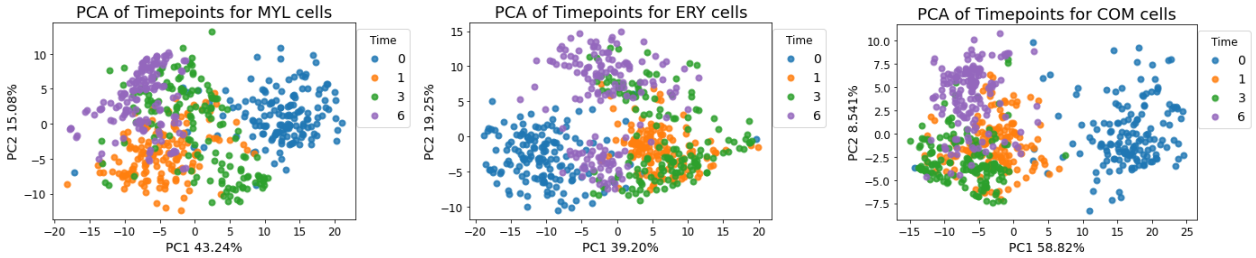


Figure 5.3: PCA projection of cells analyzed separated by treatment. Cells were separated by treatments, from left to rigth: MYL, ERY and COM. COM: cells treated with GM-CSF/IL-3 and EPO, ERY: cells treated with EPO, MYL: cells treated with GM-CSF/IL-3. Personally generated with 1_scAnalysis_BloodData.py

Figure 5.2. It can be observed for MYL on Figure 5.3, that day 1 is further from the control than day 3, and that the cells at day 6 are forming one cluster. On the contrary, for ERY cells, day 6 shows two clear clusters and is closer to the control when compared against days 1 and 3. Finally, within COM treated cells, only one cluster is formed per day, and days 1, 3 and 6 are closer between each other compared to the control cells.

5.3 ScATA applied to erythroid-myeloid-lymphoid cell differentiation data set

The scATA algorithm was applied to infer the GRN underlying the EML cell differentiation data sets. As in Section 5.2, the data sets were analyzed separately for ALL, MYL, ERY and COM. Figure 5.4 presents the ROC curve for the *improvement* and *improvement*+α_3+*rank* scores, as these scores had higher AUROC in Chapter 4. Appendix J presents the ROC curve for all the scores considered listed on Section 3.2.5. It can be observed in Figure 5.4 that scATA applied to MYL cells had a higher AUROC 0.64-0.66 than the ERY cells (0.50-0.50). ALL cells and COM cells have similar AUROC values, in between MYL and ERY cells, (0.57-0.58 and 0.59-0.60 respectively). The ROC curve for ALL cells starts with a higher true positive rate than the rest, meaning that in the first mentioned links, there are more true positives than for the rest.

An inferred GRN was constructed with the links detected by the scATA algorithm with the best 25 *Improvement* scores for each data set. These GRNs are presented below.

5.3.1 Gene regulatory network reconstruction with ALL cells time series

Figure 5.5 presents the inferred GRN when the best 25 *improvement* scores are considered. As it can be seen in the black lines (true links detected) only 5 out of the 25 links proposed were correct.

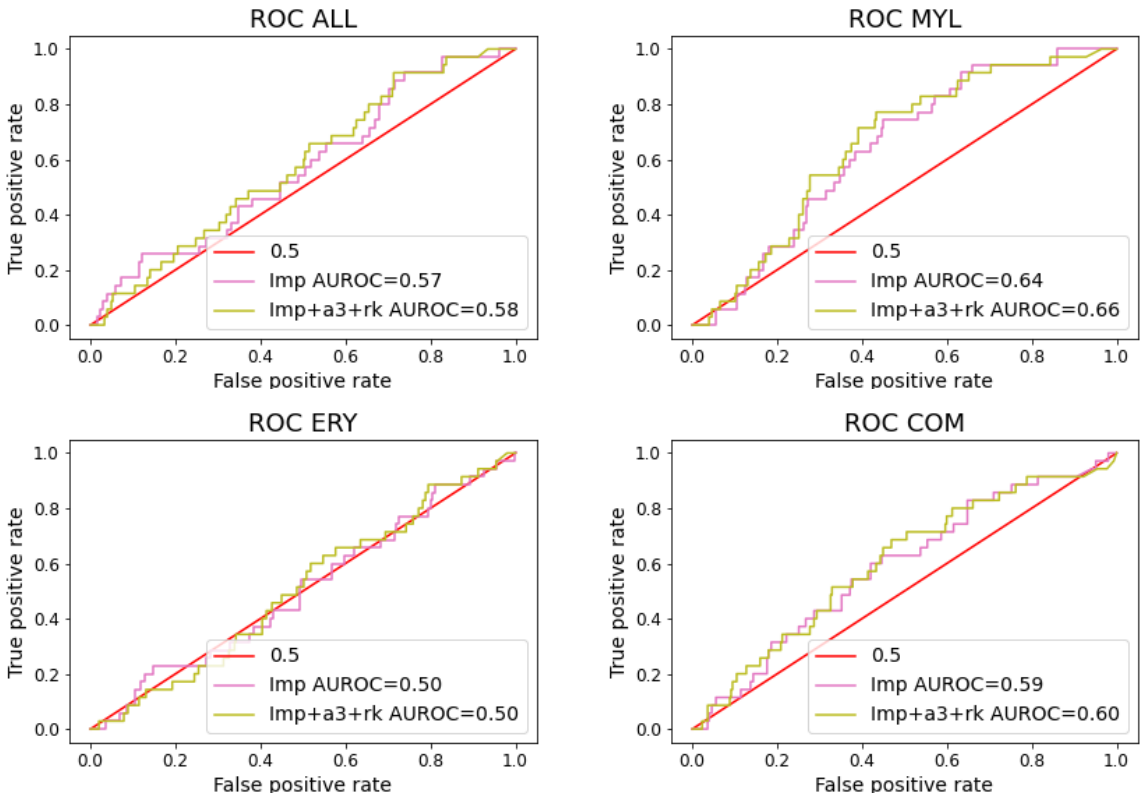


Figure 5.4: ROC curve for different time series in EML differentiation data set. ALL: all cells on the data set, MYL: cells treated with GM-CSF/IL-3, ERY: cells treated with EPO, COM: cells treated with GM-CSF/IL-3 and EPO, Imp: Improvement score, Imp+a3+rk: Improvement, Rank and Regulation Parameter α_3 score. Personally generated with 4_evaluate_scATA.py

It can also be observed in the figure that most of the link are related with gene ckit.

5.3.2 Gene regulatory network reconstruction with MYL cells time series

Figure 5.6 presents the inferred GRN when the best 25 *improvement* scores are considered. As it can be seen in the black lines (true links detected) only 2 out of the 25 links proposed were correct. It can also be observed in the figure that most of the links are related with genes CEBPa and Fog1.

5.3.3 Gene regulatory network reconstruction with ERY cells time series

Figure 5.7 presents the inferred GRN when the best 25 *improvement* scores are considered. As it can be seen in the black lines (true links detected) only 2 out of the 25 links proposed were correct. It can also be observed in the figure that most of the links are related with genes Fli1 and Gfi1.

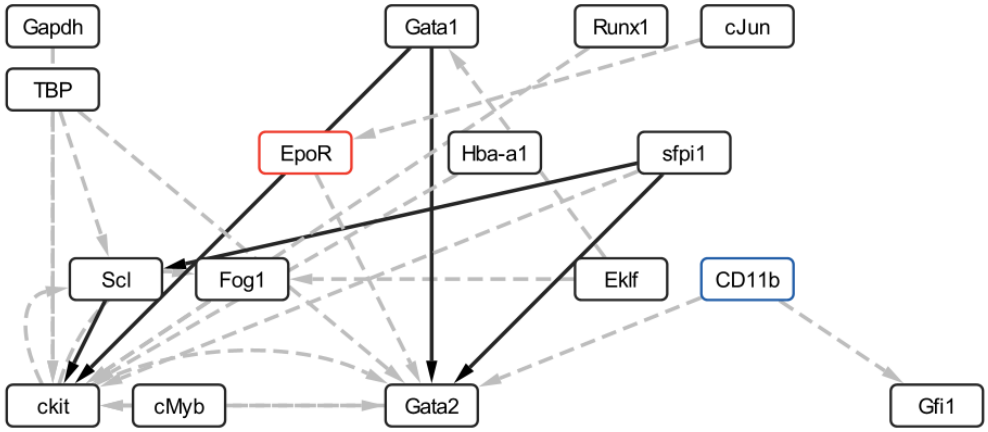


Figure 5.5: Reconstructed GRN from top 25 improvement scores for ALL cells. The 25 lines represent the 25 links detected by the best improvement scores after scATA was applied to ALL cells. Black lines are true positive links and grey lines are false positive links. The target arrow of the edge represents direction, not regulation type. Personally generated with Cytoscape.

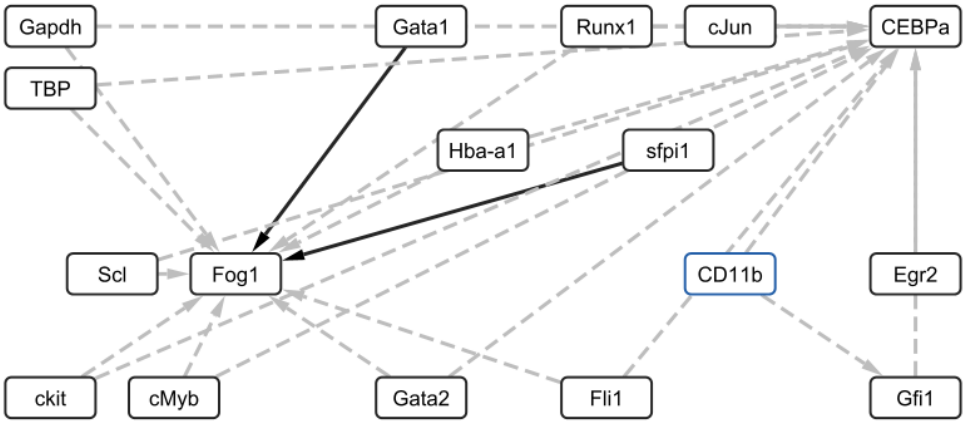


Figure 5.6: Reconstructed GRN from top 25 improvement scores for MYL cells. The 25 lines represent the 25 links detected by the best improvement scores after scATA was applied to MYL cells. Black lines are true positive links and grey lines are false positive links. The target arrow of the edge represents direction, not regulation type. Personally generated with Cytoscape.

5.3.4 Gene regulatory network reconstruction with COM cells time series

Figure 5.8 presents the inferred GRN when the best 25 *improvement* scores are considered. As it can be seen in the black lines (true links detected) only 4 out of the 25 links proposed were correct. It can also be observed in the figure that most of the links are related with genes Hba-a1 and CEBPa.

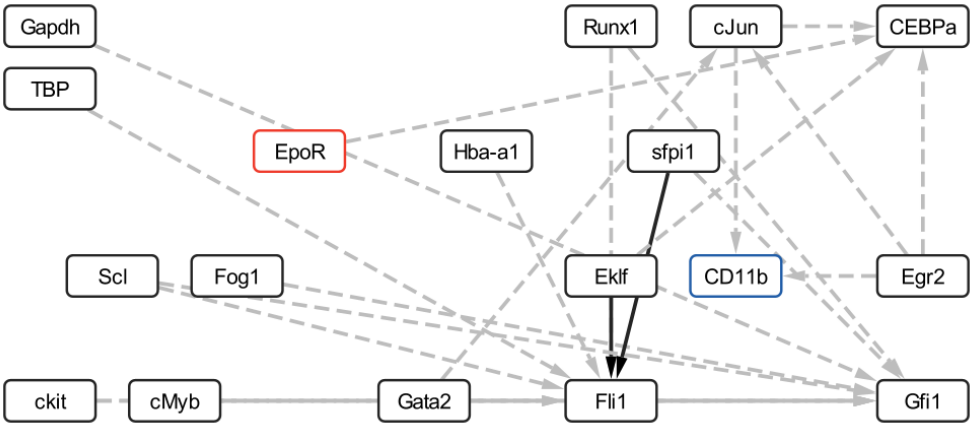


Figure 5.7: Reconstructed GRN from top 25 improvement scores for ERY cells. The 25 lines represent the 25 links detected by the best improvement scores after scATA was applied to ERY cells. Black lines are true positive links and grey lines are false positive links. The target arrow of the edge represents direction, not regulation type. Personally generated with Cytoscape.

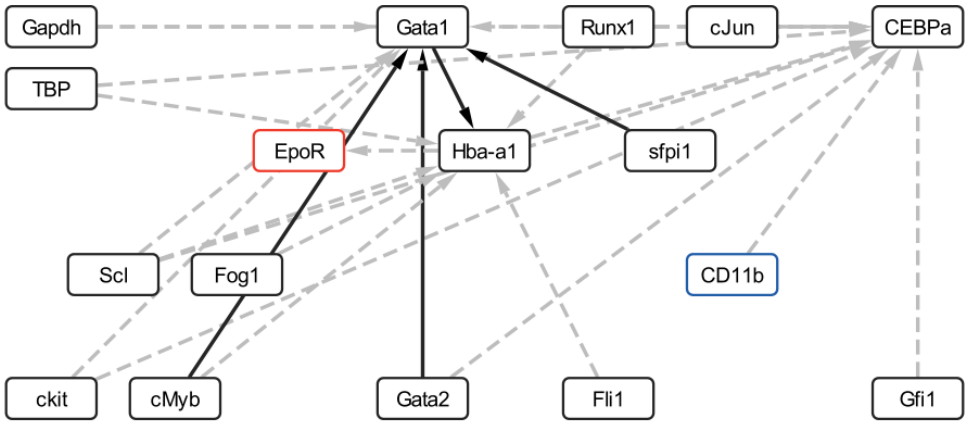


Figure 5.8: Reconstructed GRN from top 25 improvement scores for COM cells. The 25 lines represent the 25 links detected by the best improvement scores after scATA was applied to COM cells. Black lines are true positive links and grey lines are false positive links. The target arrow of the edge represents direction, not regulation type. Personally generated with Cytoscape.

Chapter 6

Discussion

The principal objective of this thesis was to develop a method to infer GRNs from single-cell transcriptomic time series data sets based on populations distributions. The key features of the desired algorithm were to be based on dynamical models, to be simple and to be scalable. With this purpose, a method to infer regulation from one gene to another (as a pair) was developed. The algorithm is based on estimating the parameters of the CLE for a linear model that, by simulating it several times, can generate an expression distribution resembling that of the investigated gene at the different time points studied. Finally, a scheme was developed where the algorithm is applied pairwise for each possible pair of genes in the data set, and returns a ranked list of the possible links.

In the first section of this chapter, the results obtained for the selection and implementation of analytical and numerical approaches, subsequently used to develop the final scATA algorithm, and the results of the algorithm developed here applied to synthetically generated and real data will be discussed. Then, for the second section of the chapter, the discussion will focus on the algorithm developed, its benefits, limitations and possible further developments. Finally, a conclusion based on the results and the discussion will be presented.

6.1 Discussion of method's development and results

In this section, the results obtained in Chapters 2, 3, 4 and 5 will be discussed. This discussion will mostly be organized based on the objectives mentioned in Section 1.7, but will also include a discussion of the numerical studies performed used as base to build the algorithm.

6.1.1 Selection of regulation model class

Section 2.1 describes the model class chosen to build the scATA method to infer GRNs. Even though there are more complex gene regulation models, as described in Section 1.6.1, the model from Figure 2.1 was chosen because of its simplicity. In the chosen model class, only one gene regulates the other gene, through a linear function. This model has already been proved in literature to be able to model gene regulatory dynamics [19], it is computationally faster to simulate several times than more complex, non-linear models, and it has less parameters to fit than other potential models. These qualities make the optimization problem easier to tackle. Thus, the simplicity of the model simplicity is key to develop an method which can be scaled to a large number of interactions.

Nevertheless, gene repression can not be modeled by this linear model class as a function of decrease in the production rate, as in the inhibition Hill function [46]. To solve this issue, during the performed simulations, inhibition is modeled as another degradation function, by allowing the parameter α_3 to be negative. As the CLE, described in Eq. (1.8) does not allow the propensities to be negative, the negative sign has to be on the state change vector. Even though the model class for inhibition worked and could identify negative regulations, other model classes that can model gene inhibition as a decrease in the production rate could be evaluated in the future.

6.1.2 Single-cell transcriptomic simulation algorithms selection

The goal of Chapter 2 was to evaluate the performance of the different single-cell simulation algorithms described in Section 1.6.2, when used to simulate the model class selected. In the rest of the thesis, simulation algorithms are used with two purposes. The first one is to use one of these single-cell simulation methods in the developed GRN inference algorithm as the dynamical formalism behind it. The motivation behind using these mathematical formalism is to be able to infer causal regulations.

The second purpose was to simulate synthetic transcriptomic time series data in which the performance of the algorithm would be tested. The benefits from using synthetic data to assess the performance of the developed methods are:

- The ground truth is known. This means that the parameters used and the topology of the network are known. Therefore, it is possible to evaluate the accuracy of the parameter estimation optimization algorithm and the GRN inference method (e.g. by answering questions such as: “how many predicted links are correct?”).

- It is possible to simulate different data sets, with different numbers of time points and dif-

ferent time separations between them, to test the performance of the algorithm under these conditions.

- Synthetic data is not affected by measurement noise as experimental data. Therefore, all the transcriptomic behaviour simulated only has the biochemical process noise.

With these two purposes in mind, Section 2.2 evaluates ODEs, the CME, the Gillespie's algorithm and the CLE. Even though ODEs do not have a stochastic component, the model class was simulated with ODEs to understand how the average of the cells population would behave (in a deterministic manner).

The CME, described in Eq. (1.3) is a representation of how the probability of the states of the system evolve through time [22]. Figure 2.5 presents this evolution through time. The CME represents a very accurate mathematical description of the evolution in time for the cell population concerning its gene expression. Nevertheless, the problem with the CME is that to obtain one set of probability distributions of the steps through time for one set of parameters, a system of N times N ODEs, being N the number of truncated maximum number of molecules, needs to be solved. This is computationally extremely demanding and, for algorithm design purposes, every step of evaluation would take too long, making the algorithm much slower than we would desire. Furthermore, the CME described in Appendix B is only for a 2-genes model, and it is even more complex to write the ODEs system for more genes.

As the CME is computationally expensive to calculate, the model class was simulated with the Gillespie's algorithm and the CLE (Figures 2.6 and 2.7). The Gillespie's algorithm implementation used was from an available python library [56] and was able to produce the simulation results presented. The Gillespie's algorithm is a very accurate simulation method when compared with the CME, but, when simulating bigger networks, it still takes a longer computing time.

In contrast, the CLE implementation as an SDE system solved by the Euler-Maruyama approximation was a personal implementation. Since the whole code for the implementation was available, it was possible to modify the design parameters of number of simulations and integration time step. To simulate trajectories of CLE with the Euler-Maruyama algorithm, two parameters are required: the number of simulations to perform and the integration time step used. In order to develop a method based on the CLE, these two parameters have to be optimally defined. On the one hand, these parameters affect the computational time of the algorithm, therefore it is preferred to have a lower number of simulations and higher integration step, to reduce the time of the simulation. On the other hand, a lower number of simulations or a larger integration step could lead to undesired errors, thus, a balance needs to be achieved. For this reason, section 2.2.4 evaluates and

compares different values for these parameters.

Figure 2.8 and Appendix D show the average trajectories for different number of trajectories simulated. In that figure, it is possible to see that an average trajectory of a small number of cells has a different trajectory when compared to the ODE trajectory (without noise), but as the number of cells simulated increases, the trajectories start to behave similar. If there were an infinite number of cells simulated, the trajectories should be the same. From Figures 2.10 and 2.9, it was concluded that 1000 cells and a time step of 1.0 have a small enough error but do not take too long to simulate.

Finally, from Figures 2.11 and 2.12 (Section 2.2.5), it was concluded that the ensemble of simulations from the Gillespie's algorithm and the numerical integration of the CLE were good approximations for the CME in this system. This agrees with the literature, where it has been demonstrated that these two methods are able to reproduce the results of the CME, respectively [48] and [49]. When compared against the ODE system, the rest of the modelling approaches (i.e. the CME, the CLE and the Gillespie's algorithm) are able to represent the intrinsic stochasticity of single-cell transcriptomics, while still providing a distribution of cells with an average consistent with the ODE. Therefore, the single molecule modeling algorithms can be used to simulate the model class with the two objectives described before: be used as mathematical formalism to develop the GRN inference algorithm and provide synthetic time series data to test the developed methods. Furthermore, both CLE and Gillespie's simulation are faster than CME, and the CLE simulation was faster than the Gillespie's simulation and the CME, so CLE was chosen for the former objective and the Gillespie's algorithm was chosen for the latter one.

6.1.3 Stochastic differential equation parameters estimation

The third specific aim of the project was to develop an optimization algorithm that can infer the parameters of the CLE, an SDE, used to describe the two gene model class. As explained before, the CLE was chose because of its accurate representation of the single-cell stochastic trajectories and because of its computational speed. As the CLE is an approximation to the CME, it simulates the trajectories of the number of mRNA molecules for each gene present on the system, for as many cells as necessary. Snapshots of the number of mRNA molecules of each gene for each cell at specific time points can be extracted from these trajectories, mimicking the time points one desires to reproduce. Finally, these snapshots can be compared with the time series data (synthetic, or from real experiments), in order to evaluate if the simulated gene expression is similar or not.

The first section of Chapter 3 describes how the optimization algorithm for parameter estimation of the CLE was built in order to approximate a time series data set obtained by simulating the model

class with the parameters from Table 2.1 by the Gillespie's algorithm. This data set is also referred throughout the chapter as the synthetic data.

The Wasserstein distance (WD), because of its definition of being a metric to measure the distance between two distributions [55], was used as the measure of difference between two distributions (simulated and synthetic). With this metric, it is possible to observe how the estimated parameters are able to mimic the synthetic data. The square root of the sum of the square WD for each gene at each time point was considered as the OF, as described on Eq. (3.1). The optimization method used to estimate the parameters was the L-BFGS-B, a quasi-Newton algorithm [60].

When the optimization algorithm described was employed to estimate the 5 parameters of the SDE system described by Eqs. (2.5) and (2.6), we faced two main challenges: 1) the parameters had different scales, and 2) the objective function had several local minima (Figure 3.2). Therefore, the rest of Section 3.1 was aimed at solving them.

The first one, the difference in the scale of the parameters, is caused by the different biological meaning the parameters have and leads to a poor performance of the numerical integration methods, causing a slow convergence to the minimum value. As it can be observed in the first column of Figure 3.2 and in Appendix F, the parameters have different scales. Inspired by [57], we solved this issue by re-scaling the parameters. For an unknown data set, the parameters are also not known, therefore a way that works for all data sets needed to be formulated. With this purpose, in the optimization algorithm, the parameters are scaled by the average value over all the cells of the number of mRNA molecules of the gene that is multiplying the parameter. As observed in the second column of Figure 3.2, the scaling of the parameters changed the shape of the OF, making the numerical optimization problem considerably more suitable to be tackled with the employed optimization algorithm.

The second challenge is that the OF evaluated had several local minima due to the presence of a stochastic term in the SDE system, representing noise in the gene expression process. Every time a simulation of the desired number of cells is performed, a random number per reaction per time point per cell is drawn from a normal distribution. This causes that every time the OF is evaluated, even with the same parameters, the result is slightly different. A solution to this problem was proposed by [58], where the noise is initialized as a previous step of the optimization algorithm, and all the OFs are evaluated with the same noise realization. When this solution was applied to the optimization algorithm, it improved the smoothness of the OF considerably (Figure 3.2), making the optimization problem of finding its global minimum much more tractable in practice.

These two solutions allowed the algorithm to estimate the same parameters as the ones used to create the synthetic data for most of the initial conditions of the parameters analyzed (Figure 3.3), improving our approach and making the optimization problem of finding the minima less difficult to tackle. Therefore, these solutions, namely scaling and noise initialisation, are applied in the scATA algorithm which we developed.

6.1.4 ScATA algorithm development

The aim of an ATA approach for GRN inference, all ready developed in our group for bulk transcriptomic time series data [19], is to develop a simple enough model that can be applied for all possible pairs of gene combinations, and then classify the performance of each of those pairs to decide if there is regulation or not. Hence, the second section of Chapter 3 uses the optimization algorithm developed in the first section, to build the scATA algorithm.

The first difference from the scATA when compared with the SDE parameter identification section, is that this algorithm only aims to find the best possible fit for the ensemble of simulations of the TG, and the trajectories of the RG are interpolated. By the structure of the model class, the expression of the RG does not depend on the expression of the TG. Thus, the interpolation aims to find the trajectories of the RG without solving an optimization problem, decreasing the number of free parameters from 5 to 3. Therefore, we have considerably gained in terms of a less complex optimization problem to solve, and a faster time for the optimization to converge. These factors are important for our final method to be up-scalable, and able to be applied to bigger data sets to infer their gene interactions.

As the optimization method employed in our scATA algorithm was not able to identify the global minimum for each of the tested parameter initial conditions, but only for most of them, it was decided that several initial conditions would be tested. After the optimization is run for different initial conditions, the best set of parameters identified (i.e. the ones that give the smallest objective function) are stored by the algorithm. This causes the algorithm to perform the optimization multiple times for the same problem (once for every initial condition), and therefore adds computing time to the final scATA algorithm. However, in the way we have implemented the algorithm, these optimization problems can be analyzed in parallel, thus exploiting the multiple processors of modern computers, which considerably reduces the final computing time.

A final consideration to be taken into account from the algorithm is that to build the scoring system, it analyzes two possible conditions: when there is, and when there is not a regulation from the RG to the TG, and then it compares them. The performance of the different scores on synthetic

data will be discussed in the next subsection.

6.1.5 ScATA algorithm performance evaluation on synthetic data

The aim of our scATA algorithm is to infer the underlying GRNs from single-cell transcriptomic time series data. Therefore, to evaluate the performance of our method, we employed it to infer the GRNs from the synthetic data generated in section 2.3.2, where the ground truth is known (because it was generated by us). It is preferred to test methods first on synthetic data because the experimental aspects, like the number of time points, the number of cells and the number of genes, can be chosen and changed to evaluate the performance of the algorithm. Additionally, as these are simulations, it is possible to simulate several different networks and topologies, and there is no experimental measurement noise, so the method can be evaluated in a cleaner setting (i.e. without measurement noise). For each of the networks evaluated, the inferred GRN was compared with the topology used to build the networks. These topologies are detailed in Appendix E.

The ROC curve, used to evaluate the performance of the algorithm throughout Chapters 4 and 5, is a standard tool to measure the inference power of these types of algorithms. The AUROC is the standard way to summarize the ROC curve and its values range from 0 to 1. An AUROC value of 1 means the algorithm is perfectly reconstructing the network, while an AUROC of 0.5 means the method is performing in the same way as if it was doing completely random choices about inferring or not a regulation (a link) between two genes. A method that has an AUROC value of 0.5 or less is considered to be performing poorly. According to [25], most of the methods presented in Section 1.4 have AUROCs between 0.44 and 0.56 when evaluated.

The simple networks evaluated in Section 4.2 (2-genes GRNs), were used only to evaluate the usability of our developed scATA algorithm. As it analyzes all the possible regulations from the RG to the TG, only 2 regulation models were analyzed for each network (gene1 regulating gene2 and gene2 regulating gene1). The scATA algorithm applied to these 2-genes networks worked, was able to estimate the parameters of the SDE function, calculated the scores for each of these regulations, and was able to give a ranking for the possible links. Therefore, it fulfills its purpose, which was to be reasonably successful in identifying the presence or absence of a regulation between a gene and another one. Because these networks were composed of only 2 genes, and there were only 2 possible links to detect, the ROC curves have a different shape than normally seen in literature, where the AUROC is most of the times between 0.5 and 1, but not exactly 1 or 0 [61] [25]. Hence, there is no need to further analyse these results, and the GRN inference power should be studied in networks with more genes, where there actually is a network of regulations, not just one.

Scores considered for method development

The different scores compared as potential classification scores to determine the presence of absence of a regulation and to rank the gene pairs based on how likely they where to have a regulation between them, listed in section 3.2.5, were evaluated in Chapter 4. During the analysis, it is possible to see how some scores perform better than others (Tables 4.2 and 4.3). From these tables, it is possible to notice that the OF score is not a good predictor. This was expected and it is not surprising, because the OF is the metric that shows how well the RG-TG model is fitting the data, but it does not contain information about if there is a regulation or not. For example, in Appendix G - Network 02_01, it can be observed that the OF score for the regulation from x_2 to x_1 is small, meaning that the simulations of CLE with the optimal parameters fit well the distribution of the synthetic data. Nevertheless, the value of α_3 is 0, meaning it fits well the data, with a final model that contains no regulation.

The absolute value of α_3 was also evaluated as a metric to decide if there is or not regulation. As it can be observed in Tables 4.2 and 4.3, this score performed better than the OF, but lower than other scores. This might be because in Eqs. (3.5) and (3.6), α_3 is multiplied to the quantity of mRNA from the RG. Therefore, if the number of mRNA molecules of the RG is high, this value might be low even if the regulation does exist. Even though this score did not classify very well the regulation links, it can be used to improve other scores, for example the OF score. If the value of α_3 is 0, it means that there is no regulation. In this new scores *OF*+α_3 and *Improvement*+α_3, the value is set very high (100) if α_3 is 0, leading the method to conclude that there is no regulation when α_3 is 0.

On the contrary, the scores which compare the OF with regulation and the OF without regulation had better performances. As it can be observed in Tables 4.2 and 4.3, the scores *difference*, *division* and *improvement* had higher AUROCs. As expected, when comparing the model with and without regulations, it is possible to notice how much adding the regulation as a possible reaction improves the model for the TG. The *division* and *improvement* scores were always the same, meaning they return the same ordered list of possible regulation links. Thus, any of these scores can be used. As mentioned before, even if there is an improvement when comparing the model with regulation and the model without regulation, this regulation might not exist. In this case, it would mean that for the optimization method without regulation, the solution is a local minimum and not the global. Therefore, this score can also be refined by applying the rule that if α_3 is 0, the score is set very high.

Assuming that, most of the times, GRNs are sparse [20], the rank score considers that each

TG is regulated by few RGs. To build the rank, the RGs of each TG are sorted by *improvement* score. Then, the RG with the lower *improvement* score is given the higher ranking, the second lower *improvement* score the second higher ranking and so forth. Then, the rank is multiplied to the *Improvement+*α_3 score, to create the *improvement+*α_3*+rank* score. Tables 4.2 and 4.3 show that, most of the times, this score is higher than the others.

After evaluating the performance of the different algorithms in synthetically generated single-cell transcriptomic time series data, there were two scores that systematically outperformed the others: *improvement* and *improvement+*α_3*+rank*. Therefore, these scores are considered for the final scATA method, are showed in the principal ROC curve figures, and are applied in Chapter 5.

Underlying GRN topologies

As it can be observed when comparing the different panels of Figures 4.2 and 4.3, the same method applied to GRNs of the same number of genes give different ROC curves and AUROC values. This allows us to hypothesise that, even though there might be other factors involved, there are some topologies that are harder for the scATA algorithm to identify than others. Appendix E presents the underlying GRNs, the trajectories followed by the cells in the space representing the numbers of mRNA molecules of the genes, and the histograms of the snapshots for each gene.

Figure 4.2 presents the ROC curves for the selected scores for each network of 5 genes. In this figure, it is possible to notice a better performance of the scATA algorithm on time series data sets with underlying GRNs 05_01, 05_02, 05_03 and 05_06. Network 05_03 included a closed loop between genes x_1, x_3 and x_5, and the method is able to identify these links, and network 05_06 contains an inhibition from x_2 to x_4, that the algorithm is able to detect. On the contrary, network 05_04 contains two regulation inputs for x_5 and has a lower AUROC. Network 05_05 also has a lower AUROC when compared with the others. From the trajectories and the snapshots of this last network, it is possible to observe that there are two genes, associated with x_2 and x_3, for which the number of mRNA molecules just decreases. This dynamic is easy to describe without regulation, and the improvement of considering the regulation of x_1 to x_3 does not significantly improve the description of this dynamic. Overall, the AUROC for GRN inference for networks of 5 genes was between 0.65 and 0.91.

Figure 4.3 presents the ROC curves from our scATA algorithm applied to infer the underlying GRN for each network of 10 genes. It is possible to notice that performance of the method is slightly lower than for 5-genes networks, but still over 0.5 in every network studied. The only network that had an AUROC value of less than 0.6 was network 10_02, which had an AUROC 0.54 (still higher

than 0.5). In the trajectories of this network (Appendix E) it is possible to observe that the dynamic is faster than the sampling time points, and that it reaches a steady state much faster than the other simulations. This causes the snapshots data used to be not as rich in dynamic information as the other data sets, which in turn leads to a lower performance of the algorithm for this network compared to the other tested. The rest of the 10-gene networks evaluated contain multiple inputs, and overall the performance of the algorithm for GRN inference of networks of 10 genes ranging between 0.6 and 0.71. In GRN inference methods benchmark studies, the AUROC values for bulk range from 0.4 to 0.8 [61] and, for the few available methods for single-cell transcriptomics, from 0.44 to 0.56 [25]. In order to compare our method with these studies, we should aim at inferring the underlying GRN from these benchmark data sets.

From both of the figures analyzed (4.2 and 4.3), it is possible to notice that most of the ROC curves have a clear increase at the beginning. This means that most of the first links proposed are true positives, and then, when the threshold for the *improvement* score is lowered, some false positives appear. This behaviour of the ATA methodology has already been observed in literature, when performing ATA in bulk transcriptomics [19].

Scalability of the scATA algorithm

The evaluation of our scATA algorithm was performed on synthetic data sets with increasing number of genes in their underlying GRN (2, 5 and 10). For each pair of possible gene combinations, excluding self-regulation, two models are studied, with and without regulation from the RG to the TG. Therefore, when analyzing a 2-genes model, 2 combinations (x_1 to x_2 and x_2 to x_1) are analyzed, and within each combination, 2 models are analyzed, studying a total of 2x2=4 models. The same analysis for 5 or 10 genes studies (5x5-5)x2=40 and (10x10-10)x2=180 models respectively. This means the algorithm evaluates (NxN-N)x2 possible models for a data set that contains N genes.

As each of these model evaluations need to solve an optimization problem repeated multiple times with different initial conditions, and because this is the part of the method which takes more time, the overall time to apply the scATA to a data set roughly increases quadratic w.r.t. the number of genes studied. The positive aspect of the ATA approach for the design of the method is that all of these (NxN-N)x2 optimization problems can be solved separately. This means that this algorithm benefits from the use of parallel computing, as each problem can be solved in a different core.

Additionally, the performance of the scATA algorithm when analyzing 10 genes does not drop dramatically, but only mildly, w.r.t. the 5 genes case. Specifically, the shape of the ROC curves,

even though the AUROC is definitely lower, remains similar to a certain extent. In both cases the links that are called first are most of the time true positives.

6.1.6 ScATA algorithm application on real data

After evaluating the performance of our scATA algorithm on synthetic data, the last aim proposed in Section 1.7 was to test this method on a real data set. For this reason, scATA was applied to a real single-cell transcriptomic data set with a publicly available ground truth, presented in [43]. There were two objectives in performing this analysis. The first one was to evaluate the usability of the scATA algorithm when applied to real data, and the second one was to see if we are able, by applying our algorithm, to reconstruct the GRN underlying the EML cell differentiation process.

Usability of scATA algorithm

The first objective, being able to apply our algorithm to real single-cell transcriptomic time series data, was successful. The scATA was able to interpolate the expression of the RG and infer the parameters of Eqs. (3.5) and (3.6) that best fit an ensemble of simulations for the TG to real transcriptomic data, for every pair of genes. Finally the scATA algorithm returned a list of ranked links for the different scores presented in section 3.2.5. As the *improvement* and the *improvement with* α_3 *and rank* scores were selected as best scoring systems in Chapter 4, they are presented in Figure 5.4. In this figure, it is possible to see that all the ROC curves for the different cell types analyzed are over the 0.5 line most of the time, and that the AUROCs are 0.5 or more, meaning it is possible to identify correctly some of the links in the initial network, most of the time better (or considerably better) than with a random classifier, and in any case never worst (Appendix J).

As mentioned in Chapter 5, to apply the scATA algorithm on the EML differentiation data, some adjustments for the number of cells in each time point were made. We expect that in future versions of the algorithm, the interpolation of the RG will allow for differences in the number of cells between time points. Despite this manual curation of the data for some time points, we applied the method directly to the publicly available data set, with no other intervention. This proves that our newly developed scATA method, based on dynamical systems to infer GRNs, can be applied to real single-cell transcriptomics data, which also has measurement noise.

Inference of the GRN underlying the EML differentiation data set

The PCAs presented in Figures 5.2 and 5.3 show how, for every one of the separated treatments, as well as for all the treatments together, there is a dynamical behaviour of the gene expression of the

cells over time. This dynamical behaviour can be observed separately for each gene in Appendix I. Hence our algorithm, designed for inferring GRN from the transcriptome of different time points, can be applied to study the biological system. If the reconstructed GRNs would perfectly reproduce the GRN described in [43] and displayed in Appendix A, the AUROC would have been 1, which was not the case. Even though the performance of our scATA method was not very high (Figure 5.4), we still used the information given by the algorithm to reconstruct the GRNs from the different data sets.

To reconstruct the GRNs the 25 links proposed when using the *Improvement* score were used. This resulted in the GRNs presented in Figures 5.5, 5.6, 5.7 and 5.8. 25 was chosen to illustrate how the GRN can be reconstructed, and as a not too high number that could be possibly validated in the laboratory by experimental techniques. As it can be observed in Figure 5.4, for this data set, the ROC curves do not have the shape described before, where the first links are always true positives. Thus, on the top 25 scoring links, there were few true positives, and they were not the top ranked.

Interestingly, the GRN reconstruction when using all the cells of the experiment at the same time has less false positives than the rest of the reconstructed GRNs. The second better GRN reconstruction was when using the cells from the combined cytokines treatment. Moreover, even though the AUROC for the GM-CSF/IL-3 treated cells (MYL) was higher than for the EPO treated cells (ERY), the reconstruction of both of these networks had only 2 true positive links. Lastly, the genes Gapdh and TBP were only present in the data set as housekeeping genes, which are not supposed to be related with the true GRN [43], but, in all of our reconstructed networks, these genes are regulating some other genes.

When analyzing the GRN inferred with the top 25 *improvement* scores when studying all cells (Figure 5.5), it is possible to notice how 10 out of the 25 links are regulating gene ckit, considered a stemness gene marker (Appendix A). ckit gene encodes for a Tyrosine-protein kinase that acts as a cell-surface marker and, when activated by stem cell factor, plays a role in hematopoiesis and stem cell maintenance [cKIT - GeneCards] [62]. As expected, the expression of this gene decreases as the differentiation process occurs (Appendix I) and, as observed when analyzing synthetic data, genes that only have a decreasing dynamic are badly explained by the scATA algorithm. A similar observation can be made when analyzing the rest of the reconstructed networks. In the three networks there are few genes that are considered to be regulated. This means that the scATA algorithm is identifying them to be regulated by other genes, but fails to say which is the actual gene regulating them.

It is interesting to compare the MYL and ERY reconstructed GRNs. In the MYL GRN (Figure 5.6), CEBPa (a TF involved in the proliferation arrest and differentiation of myeloid progenitors [CEBPA - GeneCards] [62]) and Gfi1 (an hematopoiesis transcription repressor [Gfi1 - GeneCards] [62]), both used as stemness markers, have 14 of the 25 links. The other 11 links go to Fog1, a transcriptional in erythroid cell differentiation [Fog1 - GeneCards] [62]). On the other hand, in the ERY GRN (Figure 5.7), stemness markers (CEBPa, Gfi1 and cJun) have 14 of the 25 links, and the rest are CD11b (gene that codes for a protein that, when bound to ITGB2 is implicated in white blood cell adhesive interactions [CD11b - GeneCards] [62]) and Fli1 (a DNA binding-TF [Fli1 - GeneCards] [62]), myloid associated genes. This means that the algorithm is identifying those genes that are lowering their gene expression, as regulated. Particularly, these genes belong to the other cell type studied. This might be the reason why the single-cell transcriptomic time series GRN inference algorithm is having a better performance when all the cells are analyzed at the same time.

The toggle-switch mechanism, central in this GRN [43] and also explained in [44] as the circuit that controls the EML differentiation binary cell fate decision (Gata1-sfpi1, green lines in Appendix A), is not identified in the top 25 regulatory relations from reconstructed GRNs will ALL, ERY or MYL cells. As two of these regulations are self-regulations, they will not be identified by the scATA algorithm, as the linear approach of this algorithm does not identify self-regulation. However, the reconstructed GRN from COM cells was able to detect the regulatory interaction from sfpi1 to Gata1.

6.2 Discussion of the developed scATA algorithm

Last section discussed the results obtained for each of the chapters focusing on method development and results. Throughout this section, a summary of the advantages of using our scATA algorithm, the limitations and the possible future works will be discussed. Even though there are a lot of possible improvements, the current state of the scATA algorithm allows users to infer causal interactions between genes from single-cell transcriptomic time series data. The proposed links can eventually be tested in the laboratory and might help identify entry points for future treatment development.

6.2.1 Advantages of scATA

The results obtained showed that the scATA algorithm we developed can infer GRNs from synthetically generated single-cell transcriptomic time series data with reasonably good results, comparable

to the ones described in literature. The principal advantage of this method is that it is designed to be used in single-cell data and it is not an adaptation from bulk GRN inference methods. As it uses the mathematical formalisms developed for single-molecule chemical reactions, it is able to describe the dynamics of gene expression in single-cells. These population dynamics can infer causality in the regulation and do not require a pre-ordering of the cells by a pseudotime algorithm. In fact, methods to infer GRN from bulk data are sometimes applied to single cell data after ordering them by pseudotime, however in our opinion this comes at the cost of introducing a distortion of the time information of the data due to the mapping performed by the pseudotime methods, which cannot be accounted for since the mapping between real development time and pseudotime is not evident.

From the usability perspective, the scATA algorithm solves many parameter estimation optimization problems to infer the regulatory relationship between genes. The problems analyzed are independent and can be solved separately. Therefore, this algorithm tremendously benefits from the use of parallel computing, and the time it takes to run the algorithm on a data set, depends on the number of cores available. The algorithm was designed in a computer with 4 cores, but then all the results presented were obtained in a workstation with 48 cores running in parallel. This indeed allowed us to verify in practice the dramatic gain in computational time when employing a large number of cores in parallel. Many GRN inference methods in literature employ models considering all genes of interest simultaneously, thus have more complex, realistic models, but do not exploit parallel computing. As a consequence, they can involve much longer computational times, and are limited to the number of genes they can handle because the size of the model grows with the number of genes, while in our case the size of the models employed is constant (two genes), thus our approach does not have an upper limit to the number of genes that can be considered. However, of course computational time increases and performance of the inference might decrease with increasing number of genes considered.

6.2.2 Limitations of scATA

As mentioned previously, the number of optimization problems solved by the scATA algorithm grows quadratic w.r.t the number of genes evaluated. This is not an issue for studying a network with 10 or even 20 genes, but it will be an issue if the method is intended for the complete human genome, with more than 20,000 genes. Nevertheless, the design of the algorithm allows it to benefit from parallel computing. During the development of this thesis we went from 4 to 48 cores to improve speed, and the method can be used in more powerful servers. Therefore, even though it still might take too much time to infer bigger GRNs, bigger systems can be studied.

Another important limitation of the algorithm is how it handles negative regulation. As the model class used in the CLE is linear, the negative regulation is similar to a degradation, and not a decrease in the production rate. Therefore, the algorithm is not able to clearly differentiate these two effects, which may be causing issues in the identification of regulatory links. Additionally, when two genes have similar expression profiles throughout the data set, the algorithm proposes both of them as possible regulators and it is not able to identify which gene is causing the effect. Finally, the method evaluates the regulation from one RG to one TG at a time, reducing the information used to infer the possible link from the whole information contained in the data set. Therefore, it might be leaving out important information, which might lead us to miss some regulatory dynamics, for example the effect when two genes regulate one TG. Especially when for the activation or inhibition of the TG, both other genes need to be present to fully describe the gene dynamics.

6.2.3 Possible future extensions

Based on the results obtained from applying our algorithm to synthetic and real data aiming to infer GRNs, we noticed that there are several features that would improve the usability and could potentially improve the GRN inference capability of the algorithm. While they might represent potential future extensions of this method, they are beyond the scope of this master thesis. These possible features are listed below:

- Improve optimization method.

  The optimization method used for parameter estimation is a local optimization algorithm, so it might end in a local minimum instead of the global minimum. The current solution for this is to try different sets of initial conditions of the parameters and select the best solution (i.e. the one with the lower value of the final OF). It would be interesting to try a global optimization algorithm or determine which initial conditions of the parameters give the best results, decreasing the number of times the algorithm evaluates the optimization problem. This decrease in number of initial conditions evaluated will also reduce the computational time of the algorithm. Furthermore, the optimal parameters of the model without regulation could be used as initial conditions for the optimization with regulation.

  Additionally, each cost function evaluation of the algorithm, as it calculates the trajectory of each cell with the new parameters, takes time, increasing the final computation time of the algorithm. Therefore, it might be interesting to consider a vectorization of the function and determine if the same results of the parameters can be achieved with a different number of

simulated cells.

- Consider different model classes.

  As mentioned previously, the current model class used by the model is linear. Even though this model class simplifies the modelling of the system and shortens the computational time of the simulations, it might not be the most accurate. Therefore, it would be interesting to evaluate a different model class, such as the Hill function. This new model class could solve the gene inhibition problem mentioned before, and maybe identify self-regulations.

  Furthermore, because some gene regulations occur in pairs, a different approach could be to try two inputs for each gene. The limitation of this new approach could be the number of combinations now generated, because each pair of genes acting as regulators would be evaluated as a model for every TG in the system.

- Improve score selection.

  As described in Chapter 4, the scores evaluated had different performances, measured by their AUROCs. In the current version of scATA, the regulatory links are proposed based on only two of these scores and the GRN reconstructions were performed using only with one of them. In the future, it might be interesting to evaluate which links are repeated across scores and see if a combination of these scores might improve the performance of the algorithm.

- Testing with other synthetically generated data sets.

  Even though the algorithm was tested on different synthetically generated data sets with different numbers of genes and different underlying GRN topologies, it could be tested even further. This testing should consider a systematic evaluation with different network's topological features (open and closed loops, multiple regulatory inputs, and different numbers of genes). This study used our own implementation of a single-cell simulation algorithm based on the Gillespie's algorithm, but, as mentioned in the introduction, there are other available single-cell transcriptomics simulation tools, which could potentially be used. The algorithm could also be tested with time series data sets that have already been used in previous studies (like [25]). With those data sets, the scATA algorithm could be benchmarked with other GRN inference algorithms, and a more direct comparison could be performed.

- Further study of real biological systems with scATA.

  The developed algorithm could be applied to different single-cell transcriptomic time series data sets. As the algorithm is based on studying the dynamical changes that occur in the

gene expression, it is important that these data sets have different time points, and that the gene expression changes over time. For example, it can be applied in the study of cell differentiation, or the response of cells to environmental perturbations. Then, the links proposed by the GRN predictions could be further validated by experimental techniques.

scATA application to synthetic data could be used to propose optimal experimental conditions, in terms of the number of time points evaluated, the time separation between them and the number of cells analyzed. Depending on the dynamics of the system evaluated, several numbers of mRNA molecules simulations could be performed, and then our algorithm could be applied to them. The accuracy of the different GRN reconstructions could be then used to determine these optimal experimental conditions.

6.3 Conclusions and final remarks

The aim of this thesis was to develop a simple and scalable method that can infer GRNs from single-cell transcriptomic time series data sets based on dynamical models. To fulfill that purpose, we developed a method called single-cell All-to-All (scATA), based on the chemical Langevin equation (CLE). The method analyzes each pair of possible combinations of genes, where all genes are tested as possible regulators for each TG analyzed. In each pair of genes evaluated, the system is modelled with the CLE, and the parameters that best fit an ensemble of simulations of this SDE to the data are estimated. The parameter estimation is performed by solving an optimization problem that uses the Wasserstein distance as metric to determine the difference between the distributions at every time point analyzed.

After developing out method and testing it on synthetically generated and real single-cell transcriptomic data, it is possible to conclude that an All-to-All approach, where all pair of genes are evaluated, can be used in single-cell data to study population dynamics. Our method represents the observed single-cell stochasticity by SDEs, and does not rely on correlation or pseudo-time. Given the AUROC values obtained on synthetic data, and the shape of the ROC curves, it is possible to conclude that the method performs reasonably well on synthetic data, and as expected for an ATA approach, where the first few proposed regulatory relationships are most of the times true positives. The performance of the method when applied to real single cell data decreased, but it still has an AUROC value over 0.5, which means that it performs better than a completely random choice. Additionally, it is able to predict the clear regulation of some genes, providing hypothesized links to be tested experimentally.

The final conclusion of this thesis is that a dynamical model ATA approach, using a simple linear modelling scheme can be used for GRN inference using single-cell transcriptomic time series data. This opens the road for further methods development in this area, not only with the CLE, but also with other dynamical mathematical formalisms behind them.

References

1. Davidson, E. H. & Peter, I. S. in *Genomic Control Process: Development and Evolution* 1–40 (Elsevier, 2015).
2. Mercatelli, D., Scalambra, L., Triboli, L., Ray, F. & Giorgi, F. M. Gene regulatory network inference resources: A practical overview. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* **1863,** 194430 (June 2020).
3. Davidson, E. H. & Peter, I. S. in *Genomic Control Process: Development and Evolution* 41–77 (Elsevier, 2015).
4. Blais, A. & Dynlacht, B. D. Constructing transcriptional regulatory networks. *Genes & Development* **19,** 1499–1511 (July 2005).
5. Bailey, T. L., Johnson, J., Grant, C. E. & Noble, W. S. The MEME Suite. *Nucleic Acids Research* **43,** W39–W49 (July 2015).
6. Kozomara, A., Birgaoanu, M. & Griffiths-Jones, S. miRBase: from microRNA sequences to function. *Nucleic Acids Research* **47,** D155–D162 (Jan. 2019).
7. Daily, K., Patel, V. R., Rigor, P., Xie, X. & Baldi, P. MotifMap: integrative genome-wide maps of regulatory motif sites for model species. *BMC Bioinformatics* **12,** 495 (Dec. 2011).
8. Chiu, T.-P. *et al.* DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* **32,** 1211–1213 (Apr. 2016).
9. Duttke, S. H., Chang, M. W., Heinz, S. & Benner, C. Identification and dynamic quantification of regulatory elements using total RNA. *Genome Research* **29,** 1836–1846 (Nov. 2019).
10. Pavesi, G. in *Network Biology* (ed Nookaew, I.) Series Title: Advances in Biochemical Engineering/Biotechnology, 1–14 (Springer International Publishing, Cham, 2016).
11. Koch, C. *et al.* Inference and Evolutionary Analysis of Genome-Scale Regulatory Networks in Large Phylogenies. *Cell Systems* **4,** 543–558.e8 (May 2017).

12. Glenwinkel, L., Wu, D., Minevich, G. & Hobert, O. TargetOrtho: A Phylogenetic Footprinting Tool to Identify Transcription Factor Targets. *Genetics* **197,** 61–76 (May 2014).

13. Matys, V. TRANSFAC(R) and its module TRANSCompel(R): transcriptional gene regulation in eukaryotes. *Nucleic Acids Research* **34,** D108–D110 (Jan. 2006).

14. Castro-Mondragon, J. A. *et al.* JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Research* **50,** D165–D173 (Jan. 2022).

15. Kanehisa, M., Furumichi, M., Sato, Y., Ishiguro-Watanabe, M. & Tanabe, M. KEGG: integrating viruses and cellular organisms. *Nucleic Acids Research* **49,** D545–D551 (Jan. 2021).

16. Goldman, M. *et al. The UCSC Xena platform for public and private cancer genomics data visualization and interpretation* preprint (Cancer Biology, May 2018).

17. Aoki, Y., Okamura, Y., Ohta, H., Kinoshita, K. & Obayashi, T. ALCOdb: Gene Coexpression Database for Microalgae. *Plant and Cell Physiology* **57,** e3–e3 (Jan. 2016).

18. Margolin, A. A. *et al.* ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics* **7,** S7 (Mar. 2006).

19. Mombaerts, L. *et al.* Dynamical differential expression (DyDE) reveals the period control mechanisms of the Arabidopsis circadian oscillator. *PLOS Computational Biology* **15** (ed Nie, Q.) e1006674 (Jan. 2019).

20. Aalto, A., Viitasaari, L., Ilmonen, P., Mombaerts, L. & Gonçalves, J. Gene regulatory network inference from sparsely sampled noisy data. *Nature Communications* **11,** 3493 (Dec. 2020).

21. Kalisky, T. *et al.* A brief review of single-cell transcriptomic technologies. *Briefings in Functional Genomics* **17,** 64–76 (Jan. 2018).

22. Waldherr, S. Estimation methods for heterogeneous cell population models in systems biology. *Journal of The Royal Society Interface* **15,** 20180530 (Oct. 2018).

23. Zappia, L. & Theis, F. J. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biology* **22,** 301 (Dec. 2021).

24. Fiers, M. W. E. J. *et al.* Mapping gene regulatory networks from single-cell omics data. *Briefings in Functional Genomics* **17,** 246–254 (July 2018).

25. Nguyen, H., Tran, D., Tran, B., Pehlivan, B. & Nguyen, T. A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. *Briefings in Bioinformatics* **22,** bbaa190 (May 2021).

26. Moris, N., Pina, C. & Arias, A. M. Transition states and cell fate decisions in epigenetic landscapes. *Nature Reviews Genetics* **17,** 693–703 (Nov. 2016).

27. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. *A comparison of single-cell trajectory inference methods: towards more accurate and robust tools* preprint (Bioinformatics, Mar. 2018).

28. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32,** 381–386 (Apr. 2014).

29. Hamey, F. K. *et al.* Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proceedings of the National Academy of Sciences* **114,** 5822–5829 (June 2017).

30. Lim, C. Y. *et al.* BTR: training asynchronous Boolean models using single-cell expression data. *BMC Bioinformatics* **17,** 355 (Dec. 2016).

31. Matsumoto, H. *et al.* SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. *Bioinformatics* **33** (ed Bar-Joseph, Z.) 2314–2321 (Aug. 2017).

32. Matsumoto, H. & Kiryu, H. SCOUP: a probabilistic model based on the Ornstein–Uhlenbeck process to analyze single-cell expression data during differentiation. *BMC Bioinformatics* **17,** 232 (Dec. 2016).

33. Chan, T. E., Pallaseni, A. V., Babtie, A. C., McEwen, K. R. & Stumpf, M. P. *Empirical Bayes Meets Information Theoretical Network Reconstruction from Single Cell Data* preprint (Systems Biology, Feb. 2018).

34. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. *PLOS Computational Biology* **11** (ed Prlic, A.) e1004575 (Nov. 2015).

35. Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nature Methods* **14,** 1083–1086 (Nov. 2017).

36. Specht, A. T. & Li, J. LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. *Bioinformatics* **33,** btw729 (Dec. 2016).

37. Cao, J. *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566,** 496–502 (Feb. 2019).

38. Papili Gao, N., Ud-Dean, S. M. M., Gandrillon, O. & Gunawan, R. SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics* **34** (ed Valencia, A.) 258–266 (Jan. 2018).

39. Mimeault, M. & Batra, S. K. Concise Review: Recent Advances on the Significance of Stem Cells in Tissue Regeneration and Cancer Therapies. *STEM CELLS* **24,** 2319–2345 (Nov. 2006).

40. Hough, S. R., Laslett, A. L., Grimmond, S. B., Kolle, G. & Pera, M. F. A Continuum of Cell States Spans Pluripotency and Lineage Commitment in Human Embryonic Stem Cells. *PLoS ONE* **4** (ed Reh, T. A.) e7708 (Nov. 2009).

41. Tsai, S., Sitnicka, E. & Collins, S. Lymphohematopoietic progenitors immortalized by a retroviral vector harboring a dominant-negative retinoic acid receptor can recapitulate lymphoid, myelold, and erythrold development. *GENES & DEVELOPMENT* **8,** 12 (1994).

42. Ye, Z.-j. *et al.* Complex interactions in EML cell stimulation by stem cell factor and IL-3. *Proceedings of the National Academy of Sciences* **108,** 4882–4887 (Mar. 2011).

43. Mojtahedi, M. *et al.* Cell Fate Decision as High-Dimensional Critical State Transition. *PLOS Biology* **14,** e2000640 (Dec. 2016).

44. Andrecut, M., Halley, J. D., Winkler, D. A. & Huang, S. A General Model for Binary Cell Fate Decision Gene Circuits with Degeneracy: Indeterminacy and Switch Behavior in the Absence of Cooperativity. *PLoS ONE* **6** (ed Monk, N.) e19358 (May 2011).

45. Conrad, E. D. & Tyson, J. J. in *System Modeling in Cellular Biology : From Concepts to Nuts and Bolts* 28 (The MIT Press, 2006).

46. Santillán, M. On the Use of the Hill Functions in Mathematical Models of Gene Regulatory Networks. *Mathematical Modelling of Natural Phenomena* **3,** 85–97 (2008).

47. Munsky, B. & Khammash, M. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics* **124,** 044104 (Jan. 2006).

48. Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81,** 2340–2361 (Dec. 1977).

49. Gillespie, D. T. The chemical Langevin equation. *The Journal of Chemical Physics* **113,** 297–306 (July 2000).

50. Kloeden, P. E. & Platen, E. *Numerical Solution of Stochastic Differential Equations* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1992).

51. Schaffter, T., Marbach, D. & Floreano, D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* **27,** 2263–2270 (Aug. 2011).

52. Dibaeinia, P. & Sinha, S. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems* **11,** 252–271.e11 (Sept. 2020).

53. Cannoodt, R., Saelens, W., Deconinck, L. & Saeys, Y. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications* **12,** 3942 (Dec. 2021).

54. Dobrushin, R. L. Prescribing a System of Random Variables by Conditional Distributions. *Theory of Probability & Its Applications* **15,** 458–486 (Jan. 1970).

55. Vallender, S. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications* **18,** 784–786 (1972).

56. Abel, J. H., Drawert, B., Hellander, A. & Petzold, L. R. GillesPy: A Python Package for Stochastic Model Building and Simulation. *IEEE Life Sciences Letters* **2,** 35–38 (Sept. 2016).

57. Magni, S., Succurro, A., Skupin, A. & Ebenhöh, O. Data-driven dynamical model indicates that the heat shock response in *Chlamydomonas reinhardtii* is tailored to handle natural temperature variation. *Journal of The Royal Society Interface* **15,** 20170965 (May 2018).

58. Abdalmoaty, M. R.-H., Eriksson, O., Bereza, R., Broman, D. & Hjalmarsson, H. *Identification of Non-Linear Differential-Algebraic Equation Models with Process Disturbances* in *2021 60th IEEE Conference on Decision and Control (CDC)* (IEEE, Austin, TX, USA, Dec. 2021), 2300–2305.

59. Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* **16,** 1190–1208 (Sept. 1995).

60. Zhu, C., Byrd, R. H., Lu, P. & Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software* **23,** 550–560 (Dec. 1997).

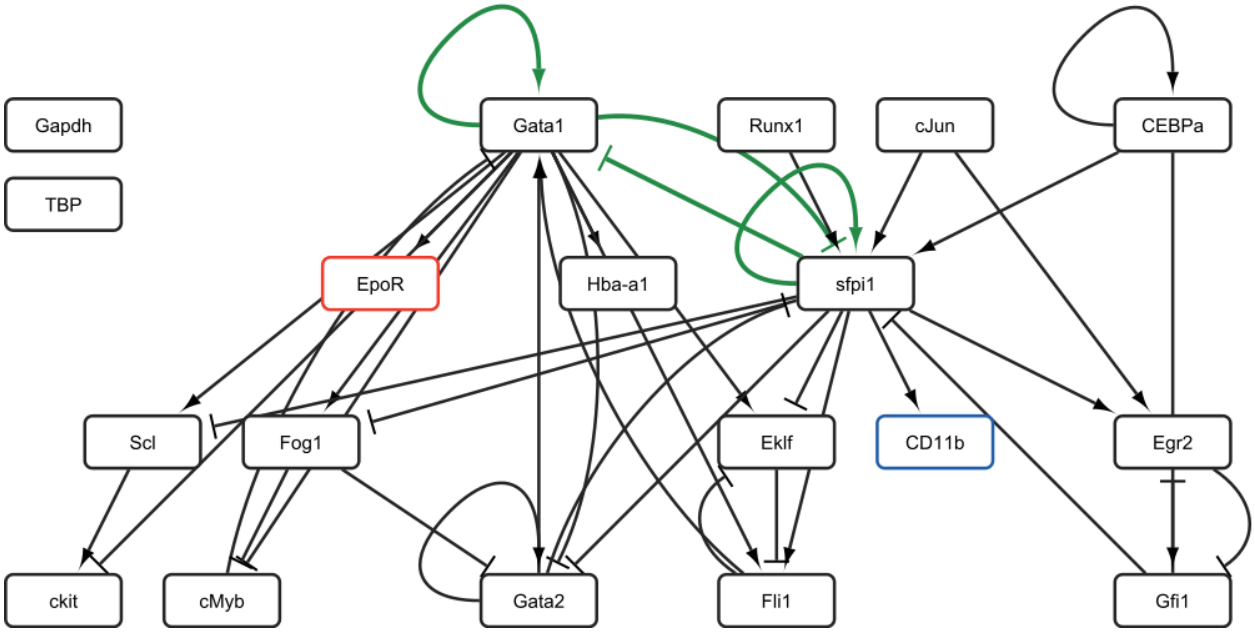
61. Mombaerts, L., Aalto, A., Markdahl, J. & Gonçalves, J. A multifactorial evaluation framework for gene regulatory network reconstruction. *IFAC-PapersOnLine* **52,** 262–268 (2019).

62. Safran, M. *et al.* in *Practical Guide to Life Science Databases* (eds Abugessaisa, I. & Kasukawa, T.) 27–56 (Springer Nature Singapore, Singapore, 2021).

Appendix

A Adaptation of GRN from EML cell differentiation and biological role of studied genes

The Figure presented has been drawn based on the underlying GRN from the differentiation process of EML cells into MYL and ERY cells [43]. Even though the GRN presented by them had 17 genes, their data set considered two more housekeeping genes, which were added to the figure, without any links to the GRN.



Underlying GRN from EML cells differentiating into MYL and ERY cells. Red box: Gene marker for ERY, Blue box: Gene marker for MYL, Green edges: Toggle switch.

The following table presents the biological associated role of each of the genes included in the GRN.

Biological associated role for genes included in EML differentiation GRN. Data obtained from [43].

Gene	Biological associated role
CEBPa	Stemness
cJun	Stemness
Egr2	Stemness
Gfi1	Stemness
sfpi1	Stemness
ckit	Stemness
CD11b	Myloid-associated
cMyb	Myloid-associated
Fli1	Myloid-associated
Gata2	Myloid-associated
Runx1	Myloid-associated
Scl	Myloid-associated
EKLF	Erythroid-associated
EpoR	Erythroid-associated
Fog1	Erythroid-associated
Gata1	Erythroid-associated
Hba-a1	Erythroid-associated
Gapdh	Control
TBP	Control

B Annotation of the CME for the model class

The following steps detail the procedure to build the CME ODEs system for the model class studied with a truncated maximum number of molecules N:

1. Write all the possible states as a vector:

$$P = \begin{bmatrix} P_{0,0} \\ P_{0,1} \\ \vdots \\ P_{1,N-1} \\ P_{1,N} \\ P_{1,0} \\ P_{1,1} \\ \vdots \\ P_{1,N-1} \\ P_{1,N} \\ \vdots \\ P_{N,0} \\ P_{N,1} \\ \vdots \\ P_{N,N-1} \\ P_{N,N} \end{bmatrix} \tag{1}$$

2. For the ordinary differential equations:

   (a) Write the equation for $\dot{P}_{0,0}$:

$$\dot{P}_{0,0} = -(\alpha_1 + \alpha_2)P_{0,0} + \beta_2 P_{0,1} + \beta_1 P_{1,0} \tag{2}$$

   (b) Write the equation for $\dot{P}_{0,x_2}$, where $x_2 \neq 0$ or N :

$$\dot{P}_{0,x_2} = -(\alpha_1 + \alpha_2 + \beta_2 x_2)P_{0,x_2} + \alpha_2 P_{0,x_2-1} + \beta_2(x_2+1)P_{0,x_2+1} + \beta_1 P_{1,x_2} \tag{3}$$

   (c) Write the equation for $\dot{P}_{0,N}$:

$$\dot{P}_{0,N} = -(\alpha_1 + \beta_2 N)P_{0,N} + \beta_1 P_{1,N} + \alpha_2 P_{0,N-1} \tag{4}$$

   (d) For every $x_1 \neq 0$ or N, write $\dot{P}_{x_1,0}$ as:

$$\dot{P}_{x_1,0} = -(\alpha_1 + \beta_1 x_1 + \alpha_2 + \alpha_3 x_1)P_{x_1,0} + \alpha_1 P_{x_1-1,0} + \beta_2 P_{x_1,1} + \beta_1(x_1+1)P_{x_1+1,0} \tag{5}$$

(e) For every pair of x_1 - x_2 where $x_1 \neq 0$ or N and $x_2 \neq 0$ or N, write $\dot{P}_{x_1,x_2}$ as:

$$\begin{aligned}\dot{P}_{x_1,x_2} =& -(\alpha_1 + \beta_1 x_1 + \alpha_2 + \alpha_3 x_1 + \beta_2 x_2) P_{x_1,x_2} + \alpha_1 P_{x_1-1,x_2} \\ &+ \beta_1 (x_1+1) P_{x_1+1,x_2} + (\alpha_2 + \alpha_3 x_1) P_{x_1,x_2-1} \\ &+ \beta_2 (x_2+1) P_{x_1,x_2+1}\end{aligned} \tag{6}$$

(f) For every $x_1 \neq 0$ or N, write $\dot{P}_{x_1,N}$ as:

$$\begin{aligned}\dot{P}_{x_1,N} =& -(\alpha_1 + \beta_1 x_1 + \beta_2 N) P_{x_1,N} + \alpha_1 P_{x_1-1,N} + \beta_1 (x_1+1) P_{x_1+1,N} \\ &+ (\alpha_2 + \alpha_3 x_1) P_{x_1,N-1}\end{aligned} \tag{7}$$

(g) Write the equation for $\dot{P}_{N,0}$:

$$\dot{P}_{N,0} = -(\beta_1 N + \alpha_2 + \alpha_3 N) P_{N,0} + \alpha_1 P_{N-1,0} + \beta_2 P_{N,1} \tag{8}$$

(h) Write the equation for $\dot{P}_{N,x_2}$, where $x_2 \neq 0$ or N :

$$\begin{aligned}\dot{P}_{N,x_2} =& -(\beta_1 N + \alpha_2 + \alpha_3 N + \beta_2 x_2) P_{N,x_2} + (\alpha_2 + \alpha_3 N) P_{N,x_2-1} \\ &+ \beta_2 (x_2+1) P_{N,x_2+1} + \alpha_1 P_{N-1,x_2}\end{aligned} \tag{9}$$

(i) Write the equation for $\dot{P}_{N,N}$:

$$\dot{P}_{N,N} = -(\beta_1 N + \beta_2 N) P_{N,N} + \alpha_1 P_{N-1,N} + (\alpha_2 + \alpha_3 N) P_{N,N-1} \tag{10}$$

C Evaluation of different integration step and different number of cells simulated in the SDE system

Due to the importance of the integration step (Δt) and the number of trajectories simulated in the Euler-Maruyama approximation, not only for computational speed but also for accuracy, these two parameters were evaluated. The following table shows how the computational time (Time run) and the percentage error of x_1 and x_2 change as the two parameters change. For each combination of Δt and N° of Simulations, the trajectories were simulated three times in order to build Figures 2.9 and 2.10. The following table shows the results of one of those runs.

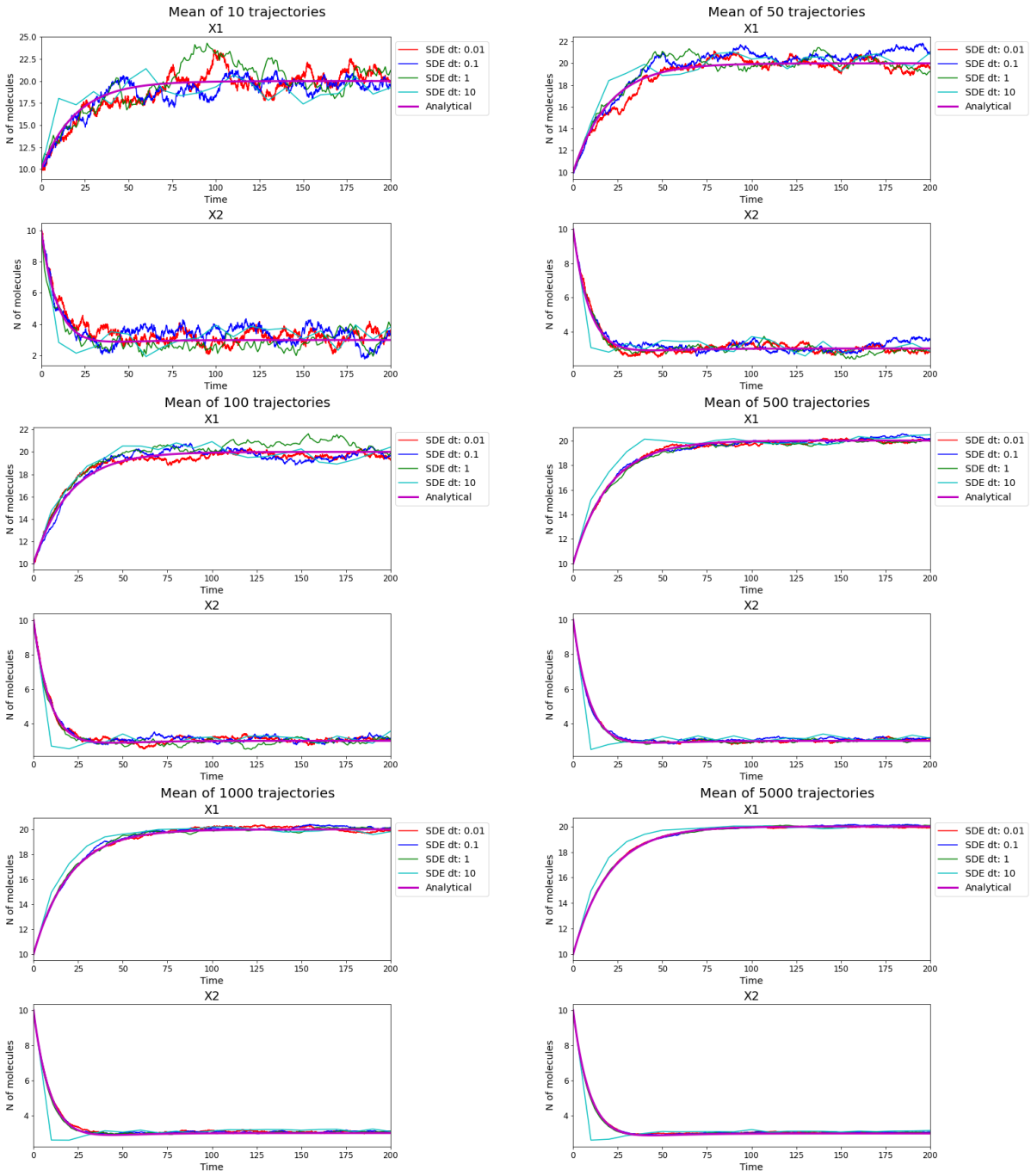
Comparison of Δt and number of simulations. Only one run (out of 3 is presented). The percentage errors are calculated with Eq. (2.7)

Run	Δt	N° of Sims.	Time Run [s]	Mean % Error (x_1 and x_2)	Mean % Error x_1	Mean % Error x_1
0	0.01	1	4.406	39.347	18.114	60.580
0	0.1	1	0.422	28.136	13.615	42.656
0	1	1	0.297	28.439	16.514	40.365
0	10	1	0.313	41.098	16.295	65.900
0	0.01	5	9.047	12.712	6.290	19.134
0	0.1	5	0.828	13.598	8.765	18.430
0	1	5	0.266	14.939	11.365	18.513
0	10	5	0.234	14.198	8.545	19.851
0	0.01	10	13.750	9.740	6.582	12.897
0	0.1	10	1.406	7.961	5.070	10.852
0	1	10	0.297	10.161	5.278	15.044
0	10	10	0.203	13.394	6.382	20.405
0	0.01	50	56.625	4.630	2.363	6.897
0	0.1	50	5.531	3.770	2.194	5.346
0	1	50	0.891	5.014	3.851	6.177
0	10	50	0.328	6.555	3.196	9.914
0	0.01	100	112.094	3.691	2.024	5.357
0	0.1	100	10.594	3.048	2.507	3.590
0	1	100	1.219	2.370	1.347	3.392
0	10	100	0.344	6.801	2.850	10.753

0	0.01	500	533.859	1.523	0.605	2.441
0	0.1	500	53.234	1.867	1.024	2.711
0	1	500	5.281	1.395	0.676	2.113
0	10	500	0.781	4.979	2.238	7.719
0	0.01	1000	1055.578	1.880	0.843	2.917
0	0.1	1000	103.656	1.322	0.559	2.085
0	1	1000	10.156	1.139	0.584	1.694
0	10	1000	1.266	5.127	1.599	8.654
0	0.01	5000	5278.969	1.214	0.233	2.195
0	0.1	5000	535.016	1.024	0.333	1.714
0	1	5000	55.906	1.015	0.346	1.684
0	10	5000	5.344	4.748	1.665	7.831

D Mean trajectory for different integration step and different number of cells simulated in the SDE system

The mean trajectory with different integration steps (Δt) and number of trajectories simulated numerically solving the CLE were evaluated and compared with the analytical solutions.



Mean trajectories for different integration time step and different number of simulated cells

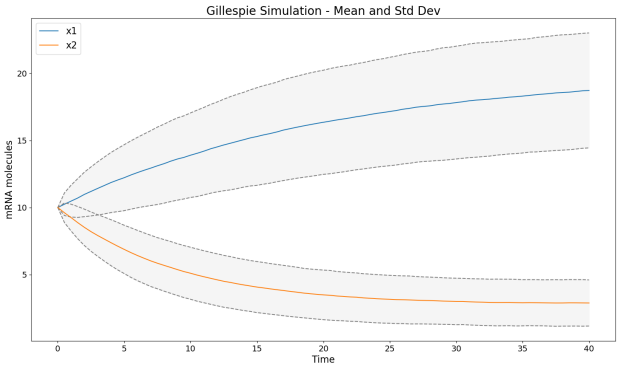
E Synthetic Networks Simulations

Description and simulation results for the different networks described in Table 2.2. This appendix separates the networks by their number of genes: 2, 5 and 10. For each network, a drawing of the topology of the network, a plot with the the mean and standard deviation of the trajectory of the simulation, and the histograms of the number of mRNA molecules per gene pare presented.

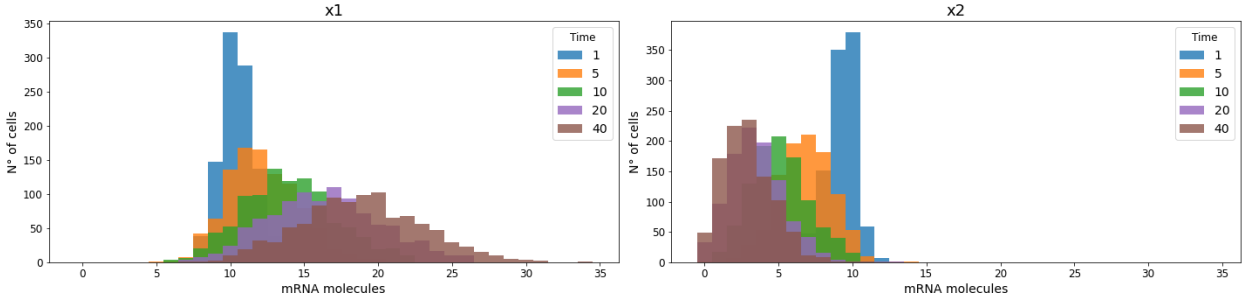
GRNs with 2 genes

Network02_01

Network drawing. Blue arrow: Activation.



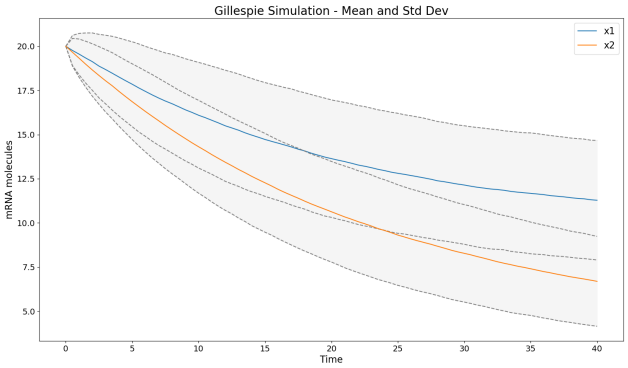
Gillespie's algorithm simulated trajectories.



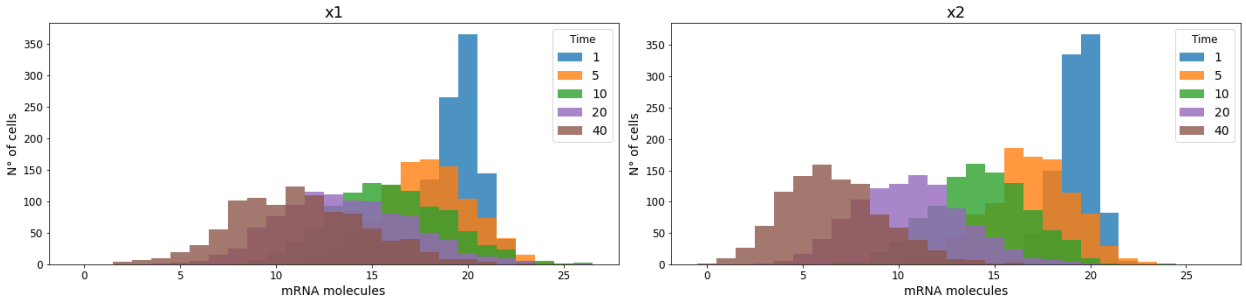
Snapshots of time points for the number of mRNA molecules for each gene.

Network02_02

Network drawing. Blue arrow: Activation.



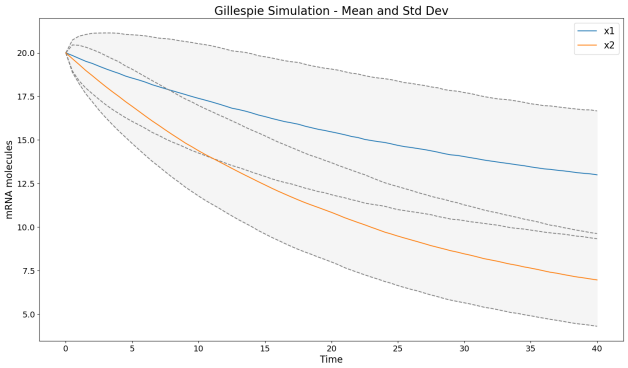
Gillespie's algorithm simulated trajectories.



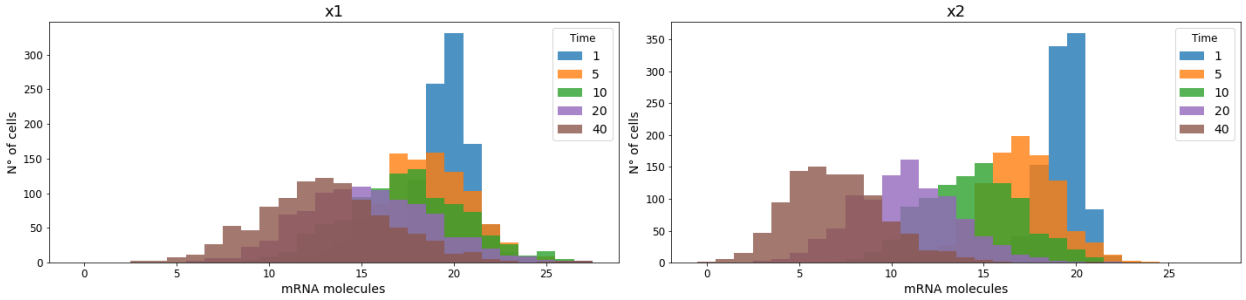
Snapshots of time points for the number of mRNA molecules for each gene.

Network02_03

Network drawing. Blue arrow: Activation.



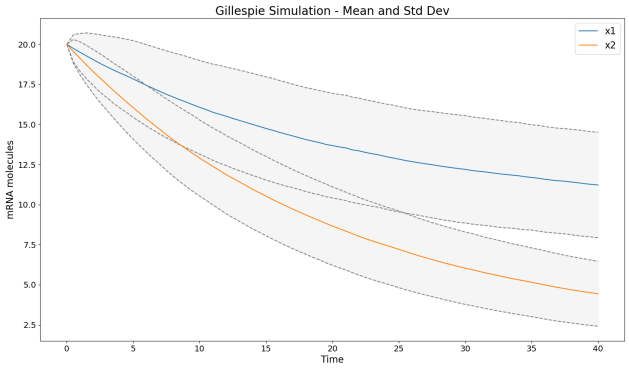
Gillespie's algorithm simulated trajectories.



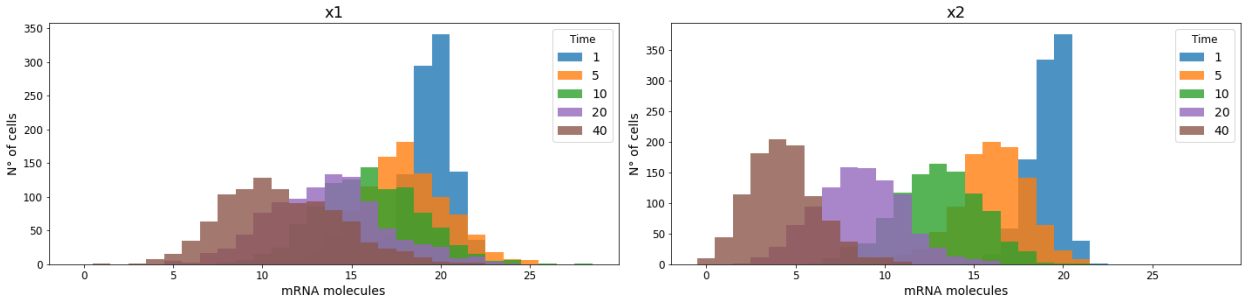
Snapshots of time points for the number of mRNA molecules for each gene.

Network02_04

Network drawing.



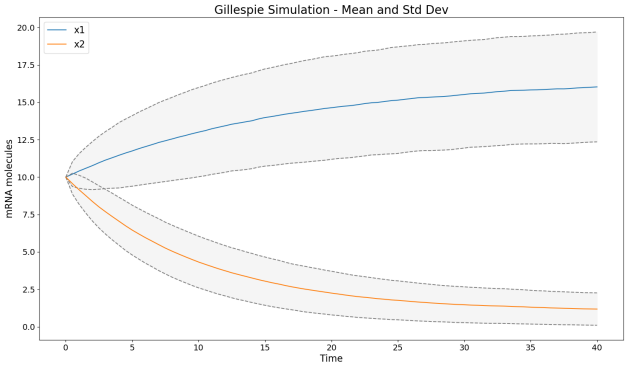
Gillespie's algorithm simulated trajectories.



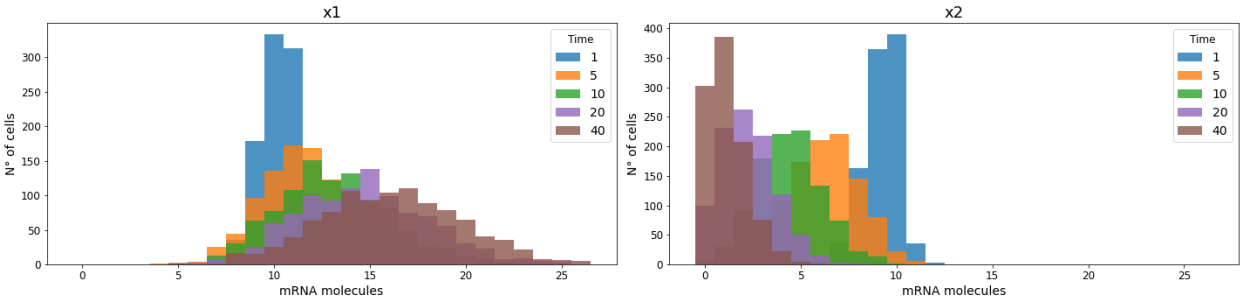
Snapshots of time points for the number of mRNA molecules for each gene.

Network02_05

Network drawing. Red Arrow: Inhibition.



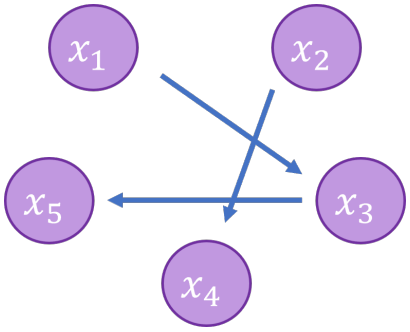
Gillespie's algorithm simulated trajectories.



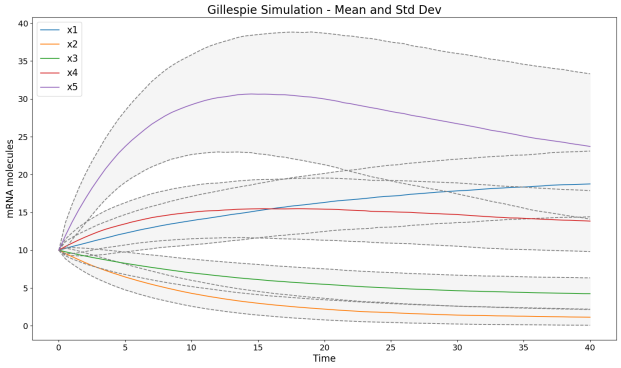
Snapshots of time points for the number of mRNA molecules for each gene.

GRNs with 5 Genes

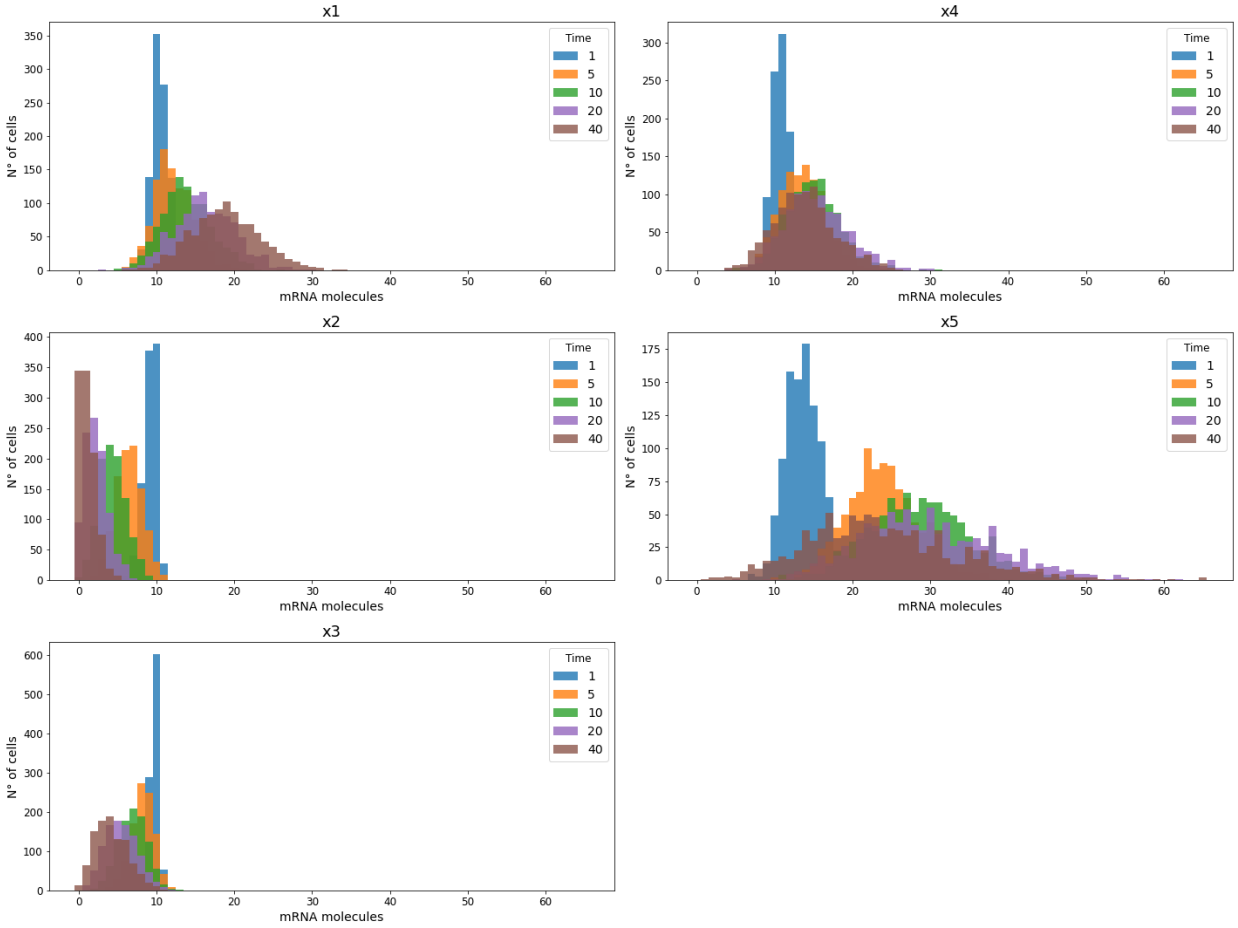
Network05_01



Network drawing. Blue arrow: Activation.

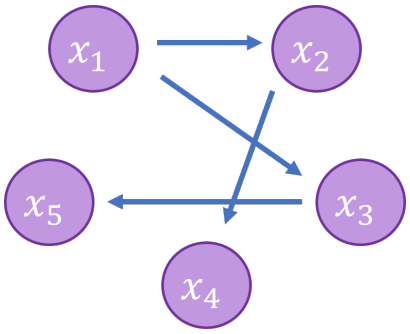


Gillespie's algorithm simulated trajectories.

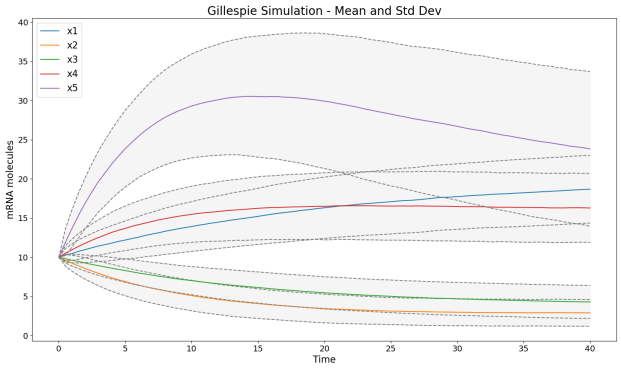


Snapshots of time points for the number of mRNA molecules for each gene.

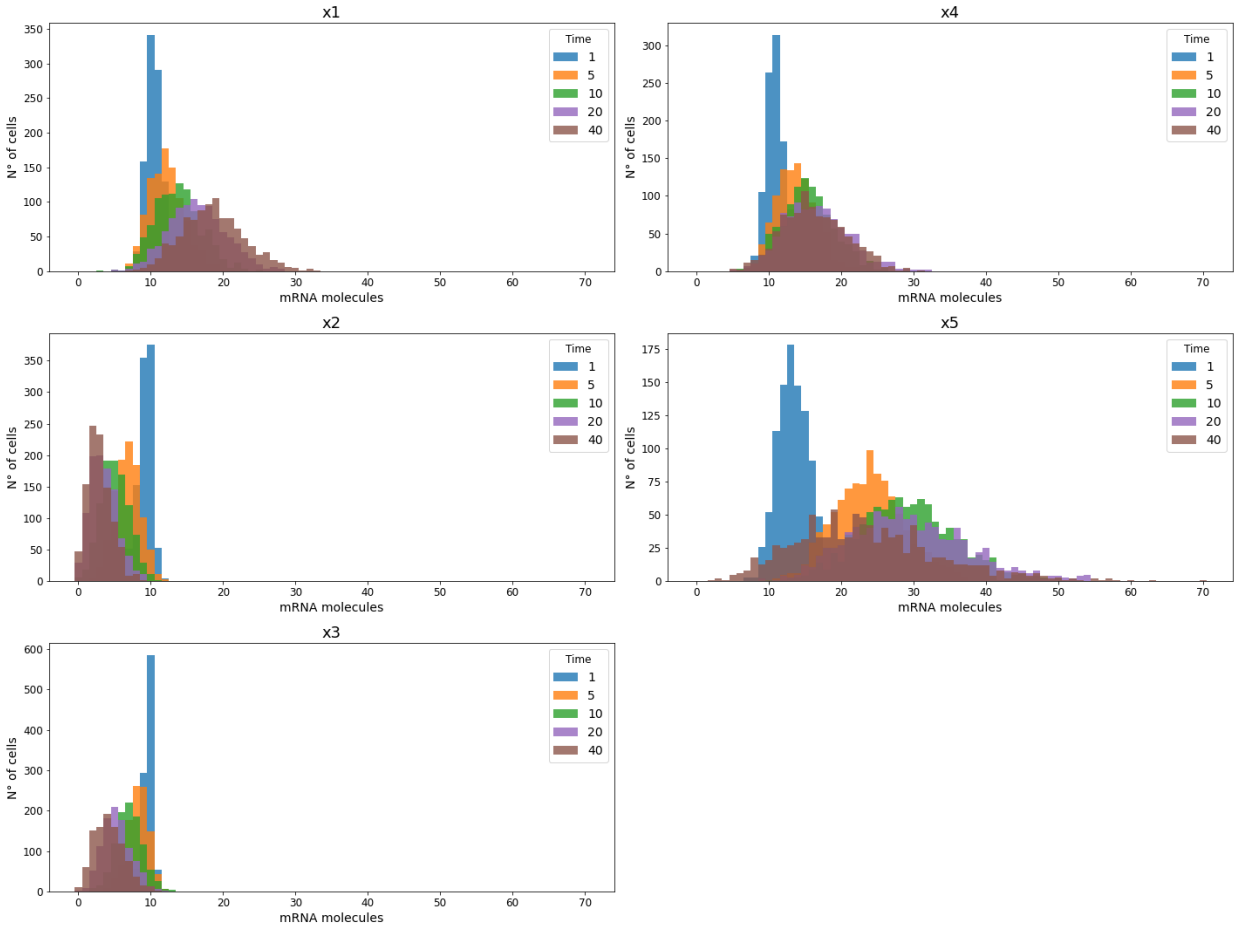
Network05_02



Network drawing. Blue arrow: Activation.

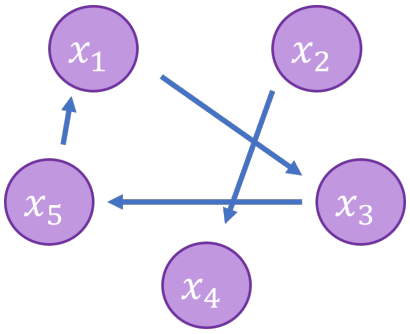


Gillespie's algorithm simulated trajectories.

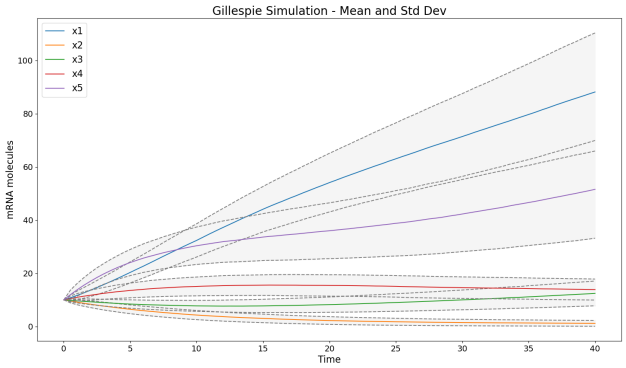


Snapshots of time points for the number of mRNA molecules for each gene.

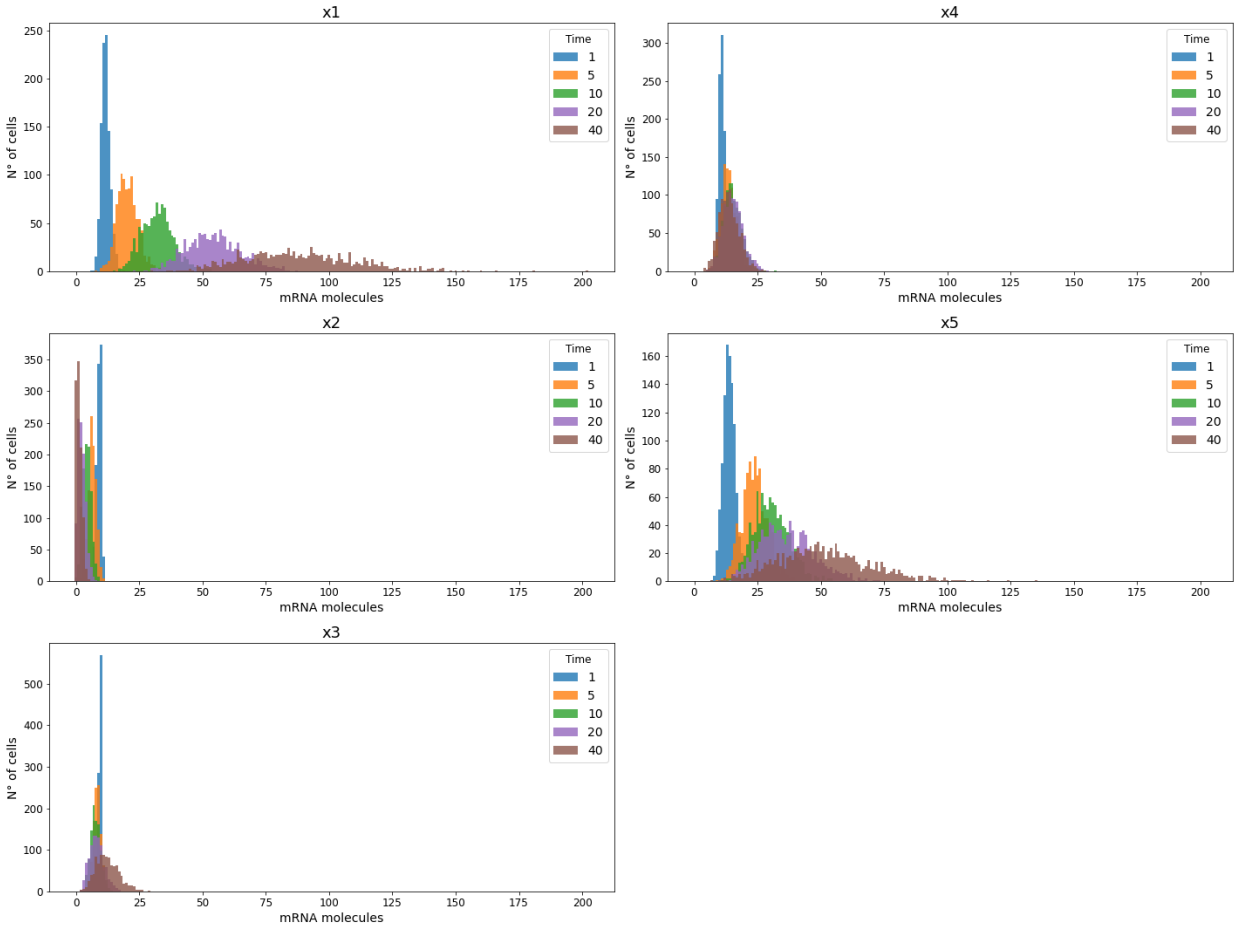
Network05_03



Network drawing. Blue arrow: Activation.

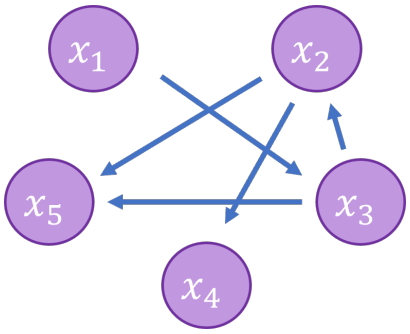


Gillespie's algorithm simulated trajectories.

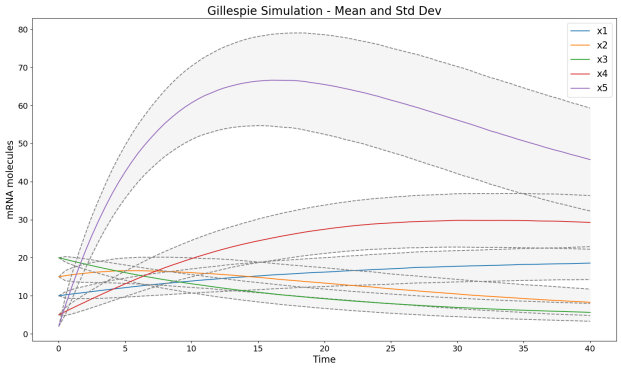


Snapshots of time points for the number of mRNA molecules for each gene.

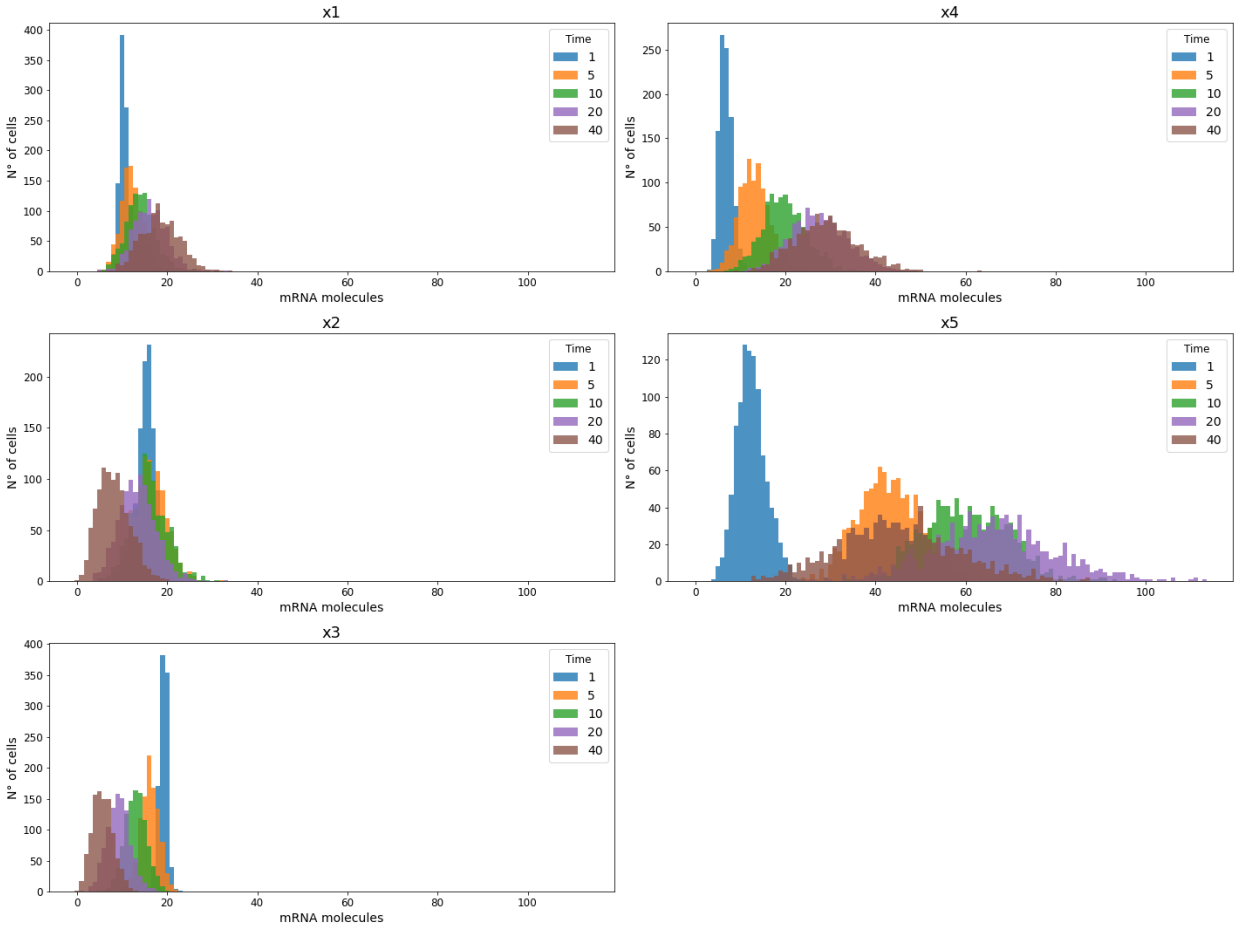
Network05_04



Network drawing. Blue arrow: Activation.

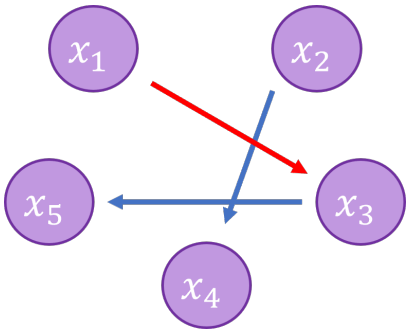


Gillespie's algorithm simulated trajectories.

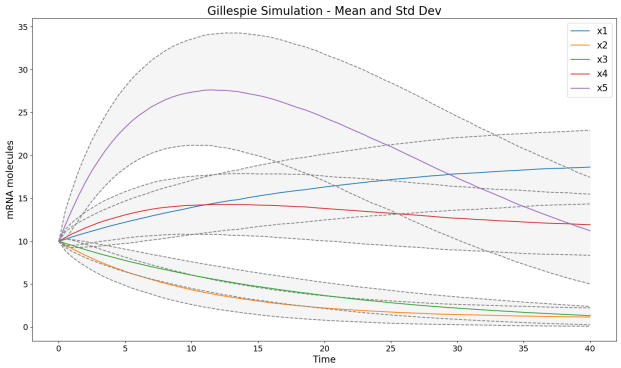


Snapshots of time points for the number of mRNA molecules for each gene.

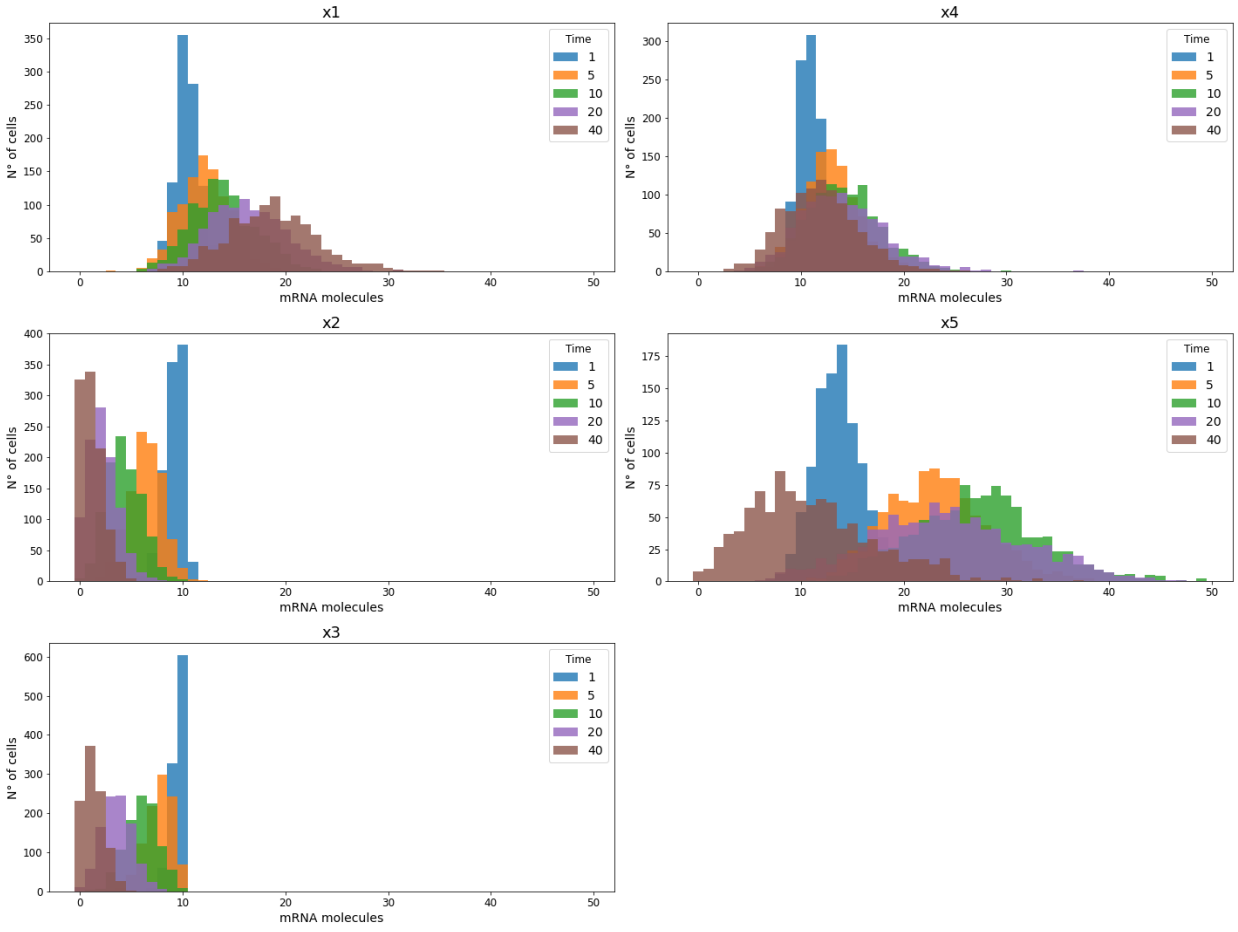
Network05_05



Network drawing. Blue arrow: Activation, Red arrow: Inhibition.

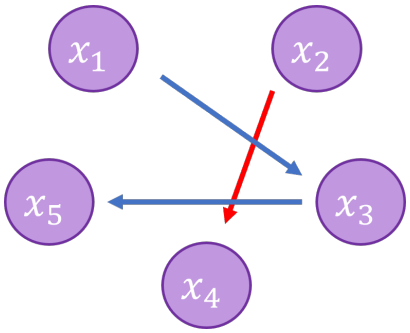


Gillespie's algorithm simulated trajectories.

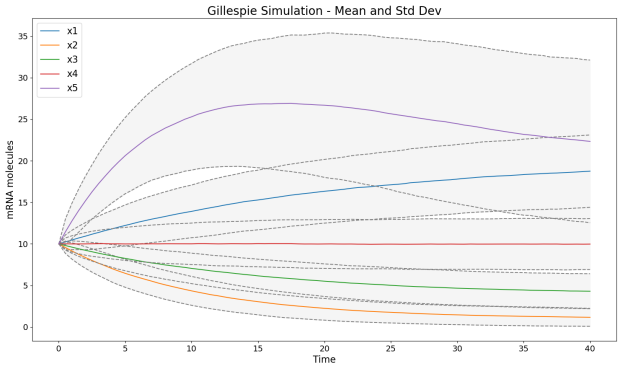


Snapshots of time points for the number of mRNA molecules for each gene.

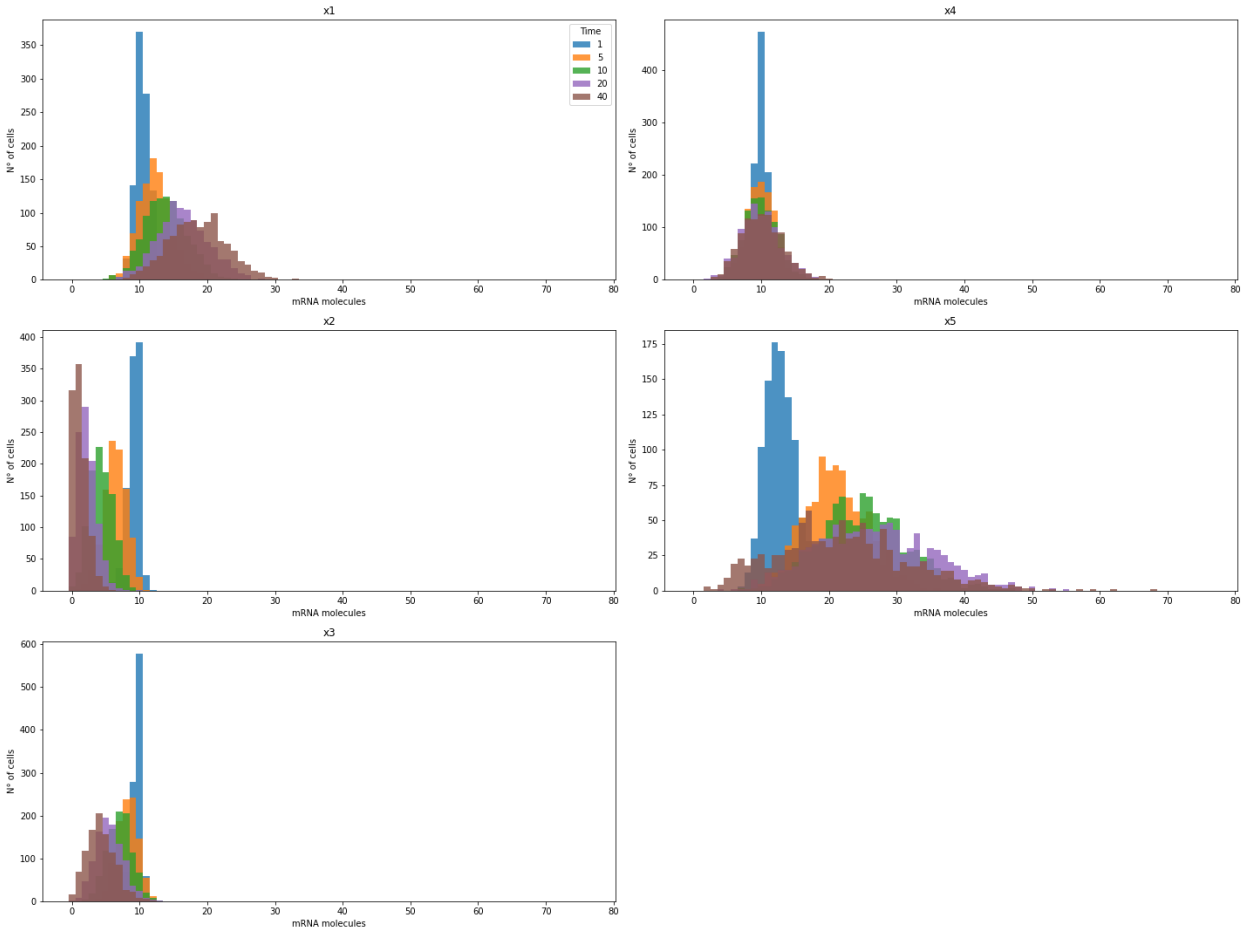
Network05_06



Network drawing. Blue arrow: Activation, Red arrow: Inhibition.



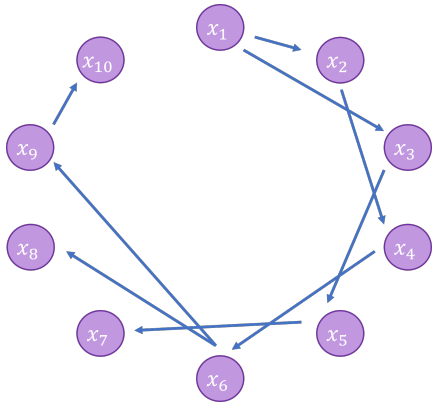
Gillespie's algorithm simulated trajectories.



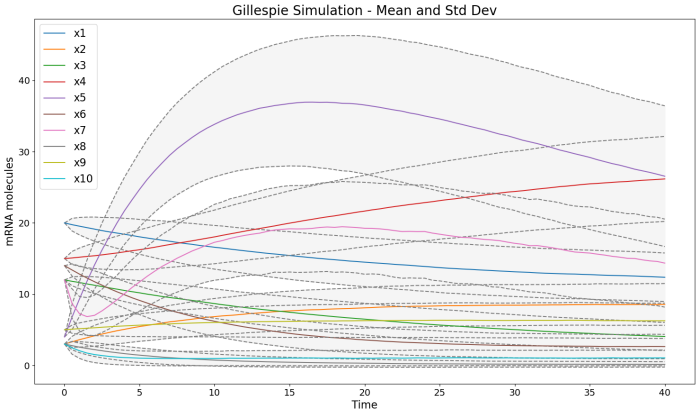
Snapshots of time points for the number of mRNA molecules for each gene.

Gene regulatory network (GRN) with 10 Genes

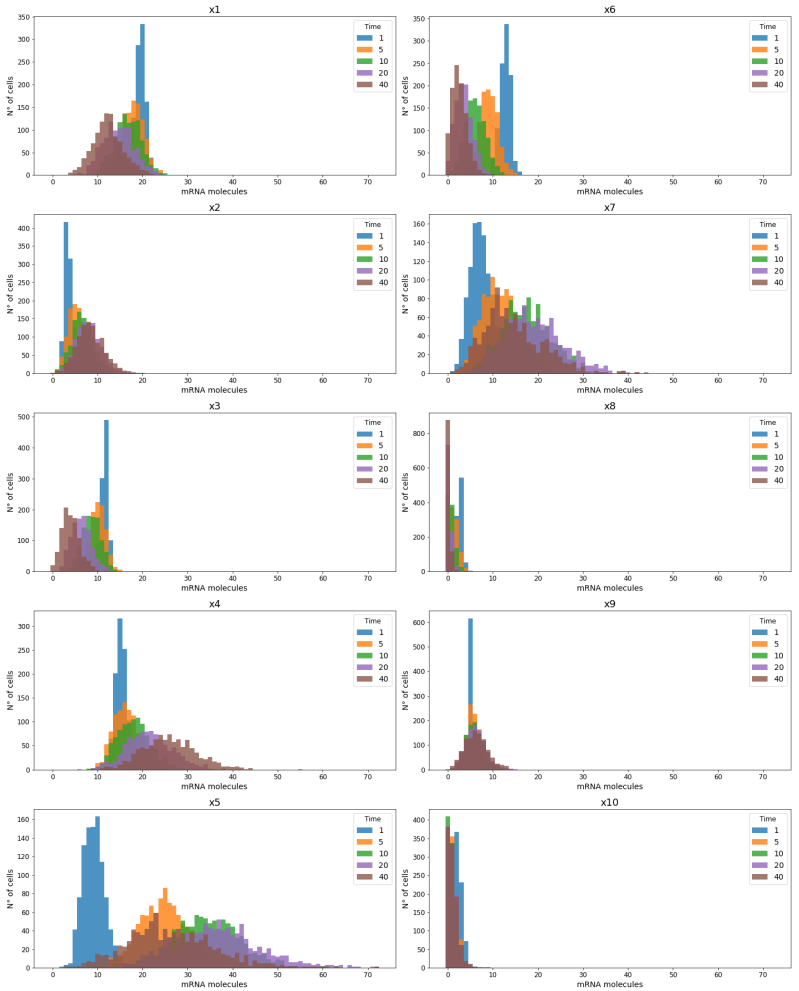
Network10_01



Network drawing. Blue arrow: Activation.

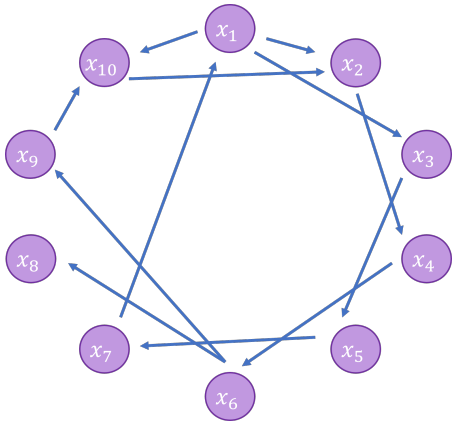


Gillespie's algorithm simulated trajectories.

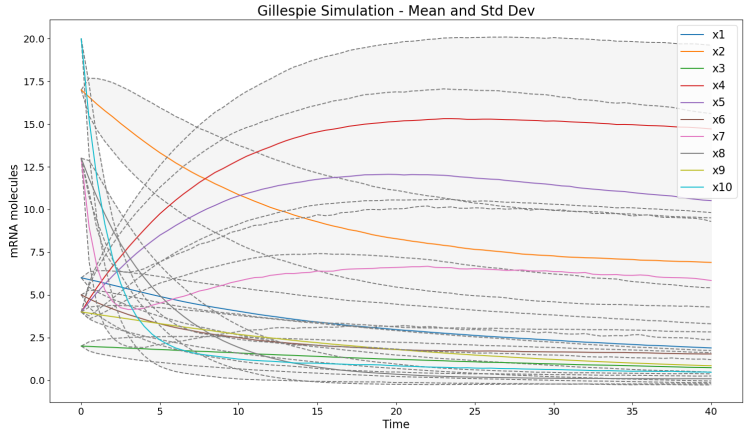


Snapshots of time points for the number of mRNA molecules for each gene.

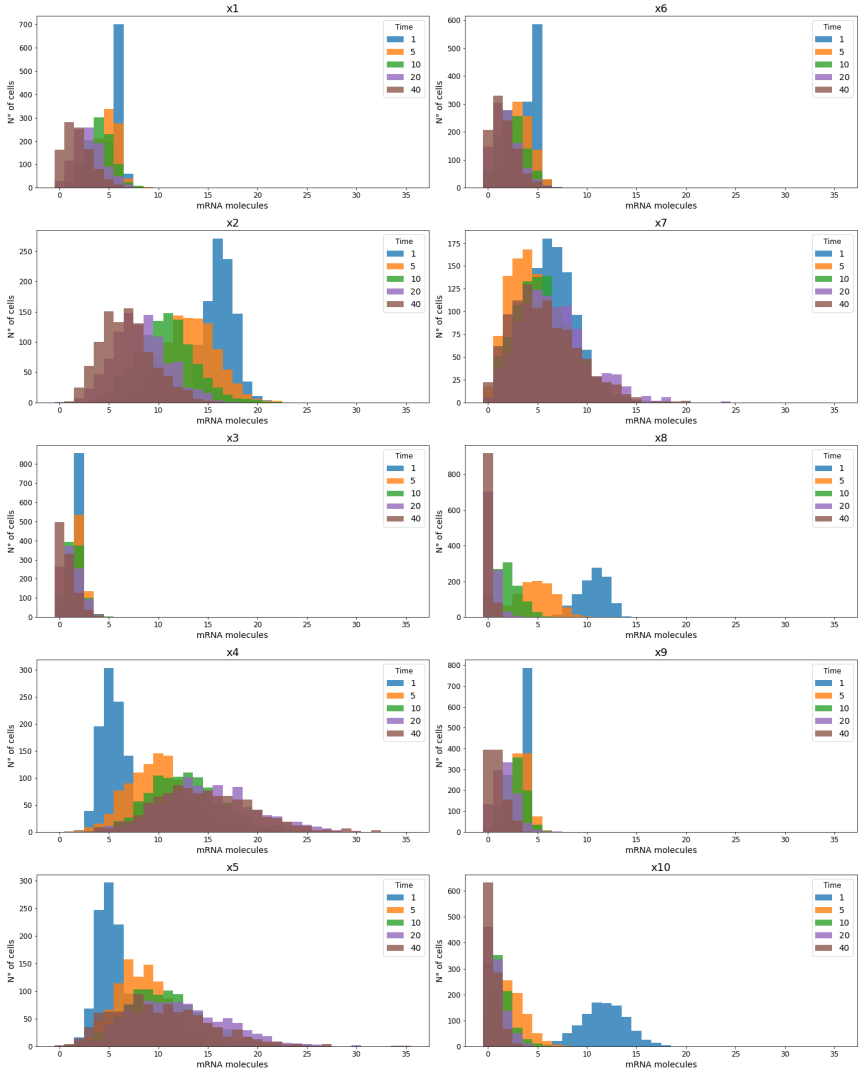
Network10_02



Network drawing. Blue arrow: Activation.

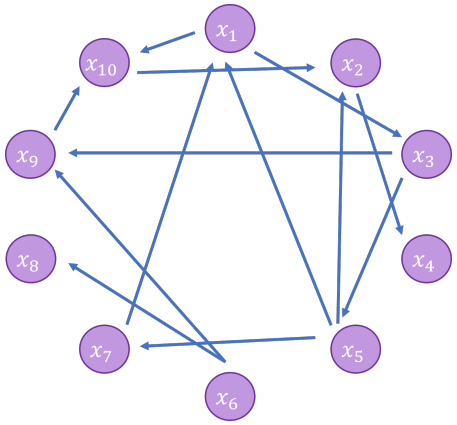


Gillespie's algorithm simulated trajectories.

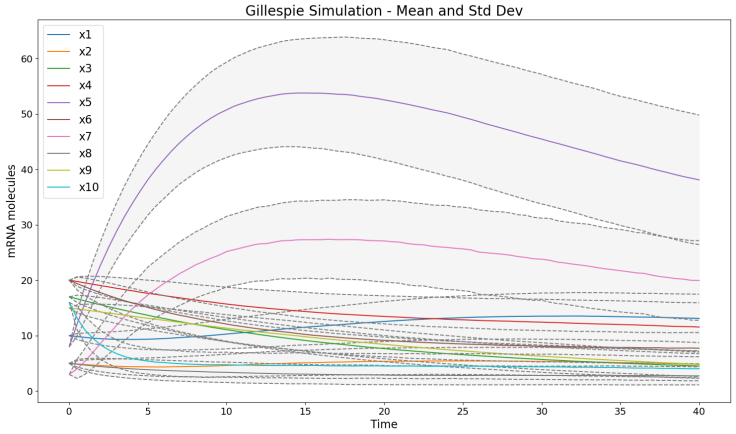


Snapshots of time points for the number of mRNA molecules for each gene.

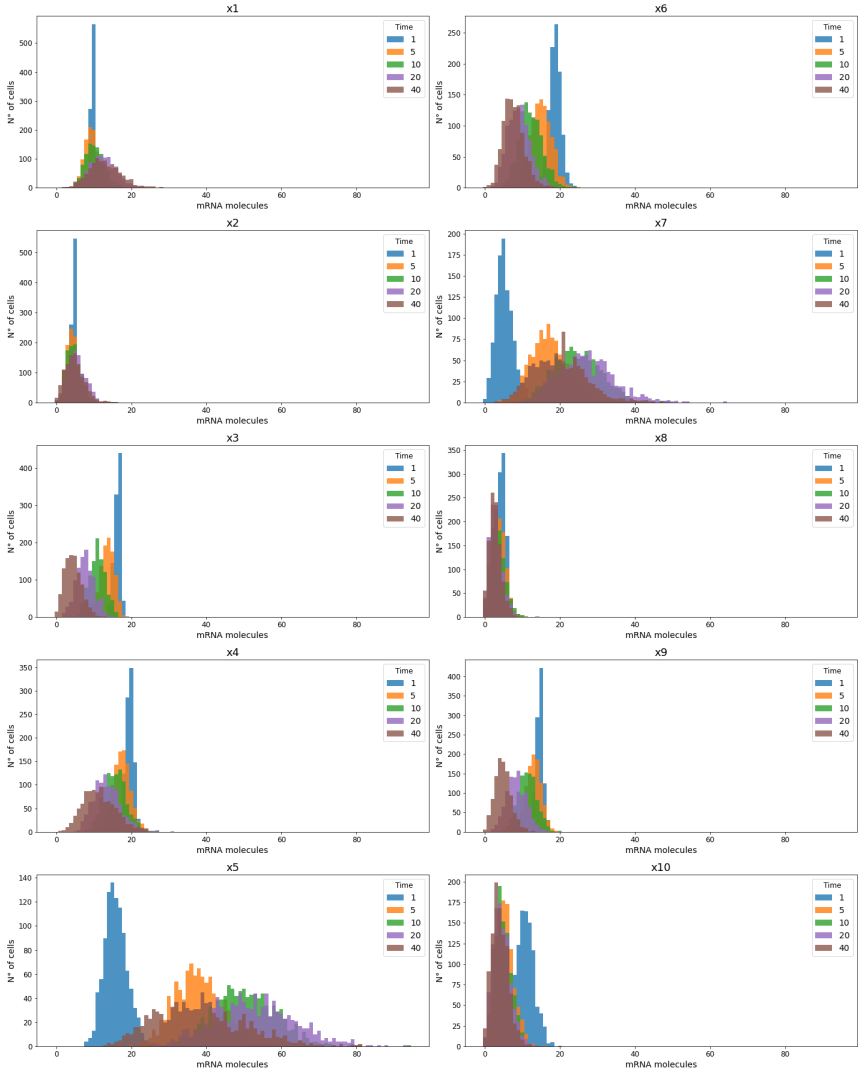
Network10_03



Network drawing. Blue arrow: Activation.

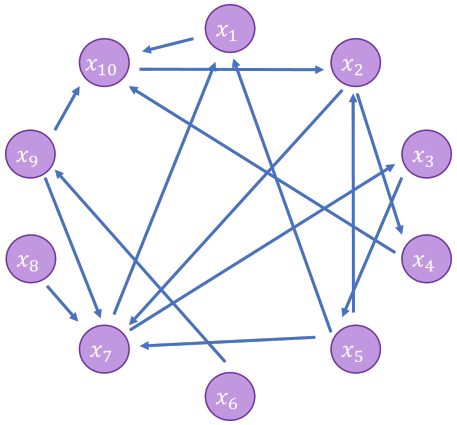


Gillespie's algorithm simulated trajectories.

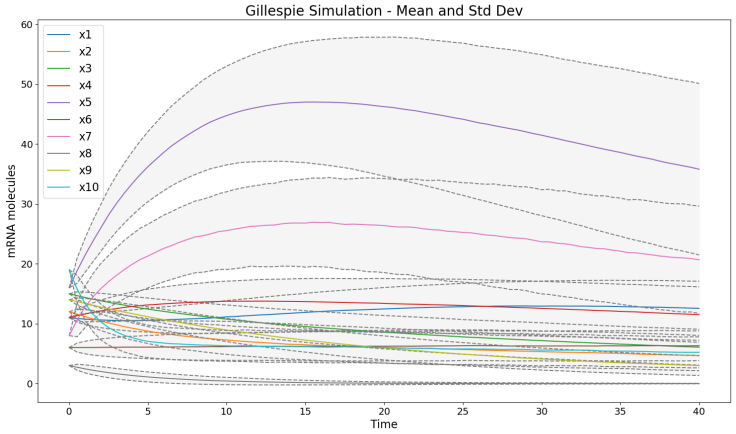


Snapshots of time points for the number of mRNA molecules for each gene.

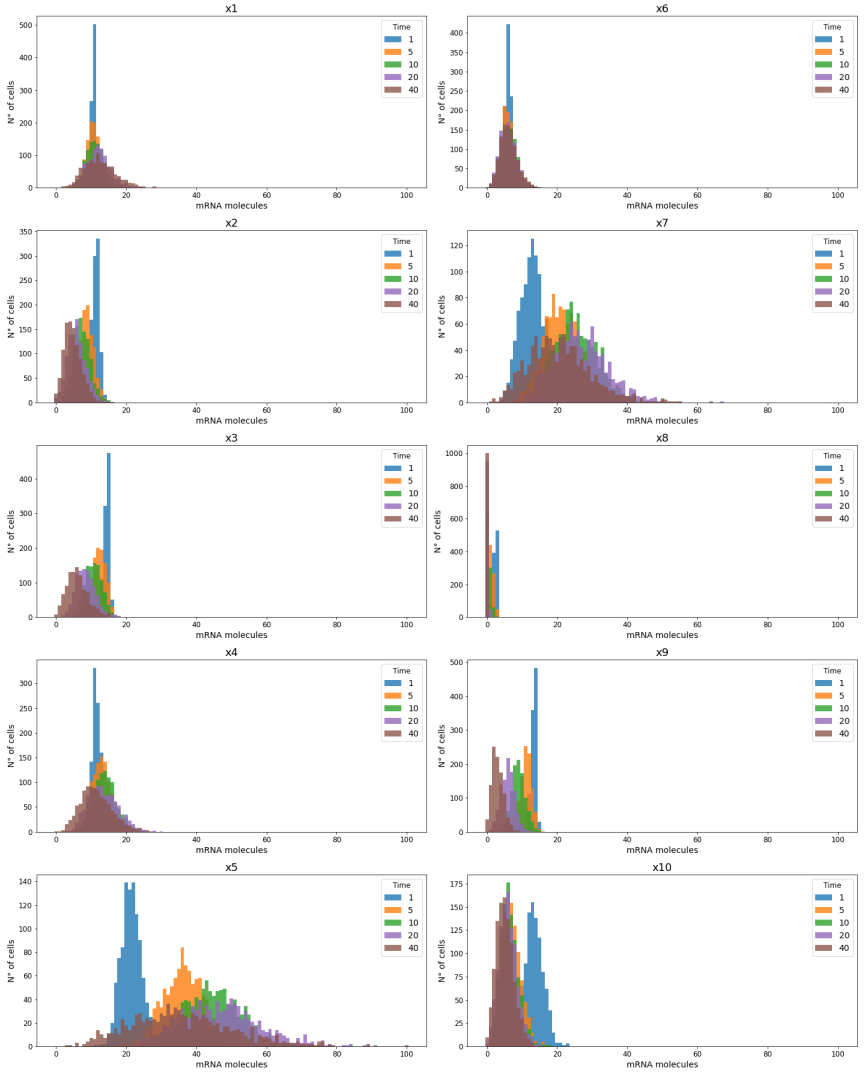
Network10_04



Network drawing. Blue arrow: Activation.



Gillespie's algorithm simulated trajectories.

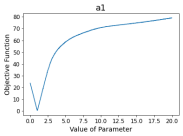
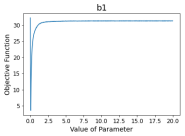
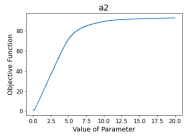
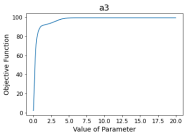
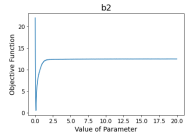
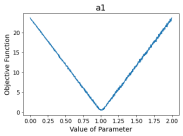
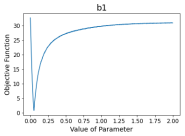
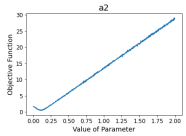
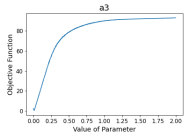
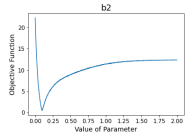
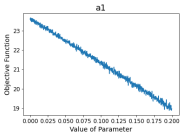
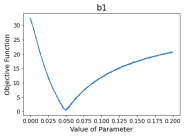
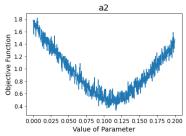
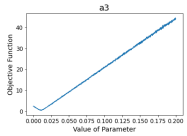
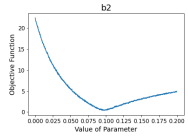


Snapshots of time points for the number of mRNA molecules for each gene.

F Objective functions evaluations

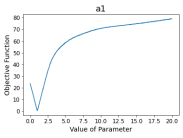
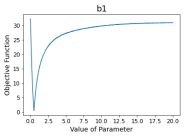
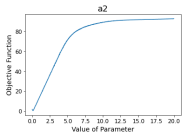
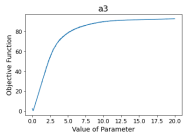
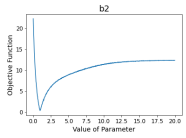
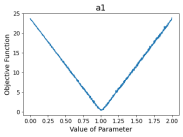
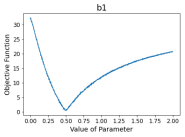
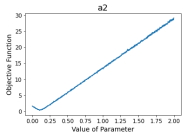
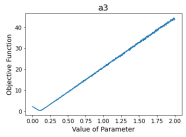
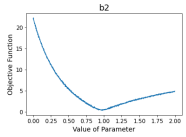
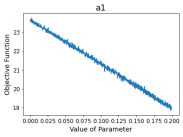
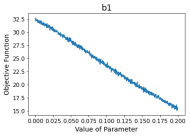
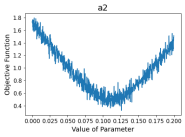
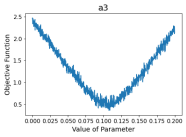
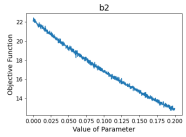
Different OFs and integration algorithms were evaluated to decide which is better (i.e. the OF and integration algorithm that, when optimized, leads to the global optimum). To determine this, the value of the OFs was calculated for each of them. For each of the figures presented below, the OF was evaluated by changing only the value of one parameter and maintaining the rest of the parameters at its optimal value (i.e. the one with which the simulation was performed). This evaluation was done for 1000 different value parameters between each of the ranges presented.

Not scaled OF and no initialized noise

Range	α_1	β_1	α_2	α_3	β_2
(0,20)					
(0,2)					
(0,0.02)					

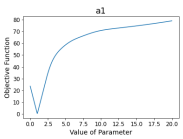
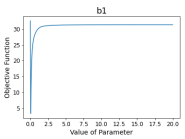
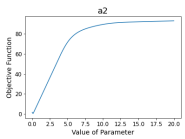
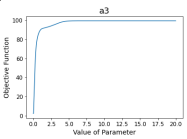
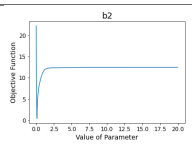
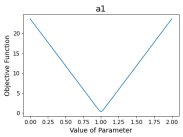
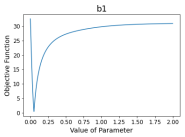
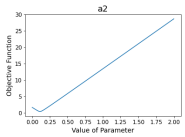
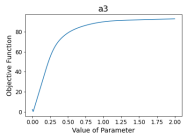
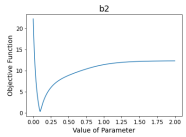
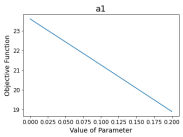
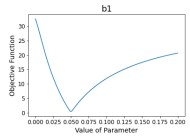
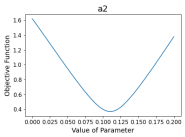
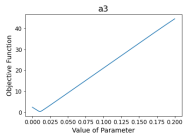
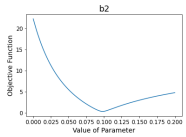
Objective function evaluation for different values of the parameters. Objective function not scaled and no noise initialization.

Scaled OF and no initialized noise

Range	α_1	β_1	α_2	α_3	β_2
(0,20)					
(0,2)					
(0,0.02)					

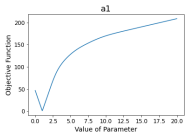
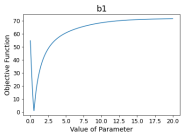
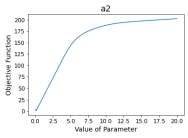
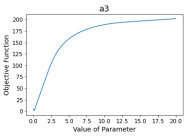
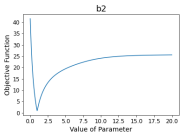
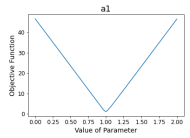
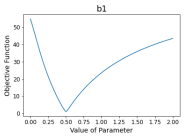
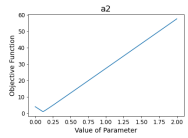
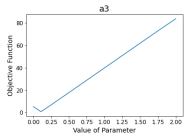
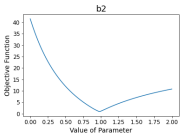
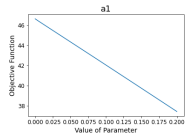
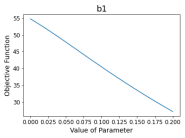
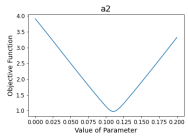
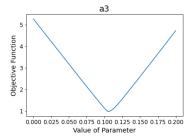
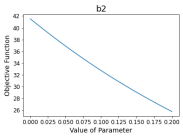
Objective function evaluation for different values of the parameters. Objective function scaled and no noise initialization.

Not scaled OF and initialized noise

Range	α_1	β_1	α_2	α_3	β_2
(0,20)					
(0,2)					
(0,0.02)					

Objective function evaluation for different values of the parameters. Objective function not scaled and noise initialization.

Scaled OF and initialized noise

Range	α_1	β_1	α_2	α_3	β_2
(0,20)					
(0,2)					
(0,0.02)					

Objective function evaluation for different values of the parameters. Objective function scaled and noise initialization.

G Example results table from scATA

Example tables with all the scores for each combination of genes for three networks of 2 genes and one network of 5 genes.

scATA result table for Network 02_01.

Regulator	Target Gene	True Link	OF Reg.	OF NotReg.	α_3	Dif.	Div.	Imp.	Rank	OF & α_3	Imp. & α_3	Imp., α_3 & Rank
x2	x1	0	0.229	0.242	0.000	-0.013	0.948	-5.2	1	100.000	100.0	100.0
x1	x2	1	0.107	0.114	0.007	-0.007	0.936	-6.4	1	0.107	-6.4	-6.4

scATA result table for Network 02_03.

Regulator	Target Gene	True Link	OF Reg.	OF NotReg.	α_3	Dif.	Div.	Imp.	Rank	OF & α_3	Imp. & α_3	Imp., α_3 & Rank
x2	x1	1	0.571	0.648	0.000	-0.077	0.881	-11.9	1	100.0	100.0	100.0
x1	x2	1	0.279	0.288	0.007	-0.009	0.968	-3.2	1	0.3	-3.2	-3.2

scATA result table for Network 02_04.

Regulator	Target Gene	True Link	OF Reg.	OF NotReg.	α_3	Dif.	Div.	Imp.	Rank	OF & α_3	Imp. & α_3	Imp., α_3 & Rank
x2	x1	0	0.340	0.333	0.000	0.007	1.021	2.1	1	100.0	100.0	100.0
x1	x2	0	0.220	0.223	0.000	-0.003	0.987	-1.3	1	100.0	100.0	100.0

scATA result table for Network 05_01.

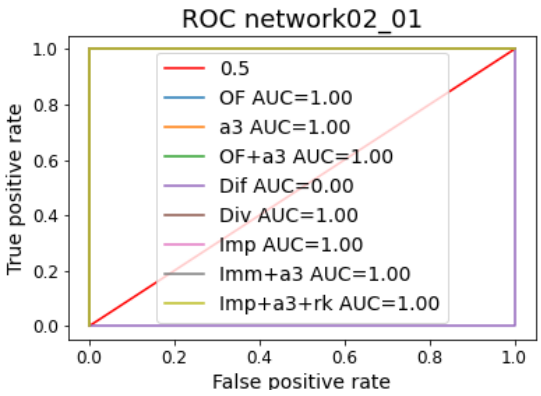
Regulator	Target Gene	True Link	OF Reg.	OF NotReg.	α_3	Dif.	Div.	Imp.	Rank	OF & α_3	Imp. & α_3	Imp., α_3 & Rank
x3	x1	0	0.24	0.41	-0.008	-0.17	0.59	-41.5	4	0.24	-41.49	-165.95
x2	x1	0	0.28	0.41	-0.008	-0.14	0.67	-32.8	3	0.28	-32.79	-98.36
x4	x1	0	0.27	0.39	-0.004	-0.13	0.68	-32.4	2	0.27	-32.44	-64.87
x5	x1	0	0.30	0.39	0.001	-0.09	0.77	-23.2	1	0.30	-23.20	-23.20
x1	x2	0	0.13	0.14	-0.001	-0.01	0.92	-7.9	4	0.13	-7.92	-31.70
x3	x2	0	0.11	0.12	0.000	-0.01	0.94	-5.8	3	0.11	-5.78	-17.34
x5	x2	0	0.11	0.12	0.000	-0.01	0.95	-4.6	2	0.11	-4.55	-9.11
x4	x2	0	0.11	0.11	0.000	0.00	0.96	-3.6	1	0.11	-3.56	-3.56
x2	x3	0	0.17	0.28	-0.013	-0.11	0.59	-40.7	4	0.17	-40.72	-162.87
x1	x3	1	0.19	0.29	0.007	-0.10	0.66	-34.4	3	0.19	-34.37	-103.11
x5	x3	0	0.27	0.29	0.004	-0.01	0.95	-4.9	2	0.27	-4.89	-9.79
x4	x3	0	0.31	0.28	0.007	0.02	1.09	8.6	1	0.31	8.56	8.56
x2	x4	1	0.34	3.00	0.082	-2.67	0.11	-88.8	4	0.34	-88.80	-355.20
x3	x4	0	0.57	3.01	0.100	-2.44	0.19	-81.0	3	0.57	-81.00	-243.01
x5	x4	0	3.00	3.01	0.000	-0.01	1.00	-0.3	2	3.00	-0.31	-0.62
x1	x4	0	3.01	3.01	0.000	0.00	1.00	0.0	1	100.00	100.00	100.00
x2	x5	0	10.34	15.59	0.109	-5.25	0.66	-33.6	4	10.34	-33.65	-134.59
x3	x5	1	10.73	15.57	0.104	-4.84	0.69	-31.1	3	10.73	-31.08	-93.23
x4	x5	0	11.85	15.58	0.089	-3.73	0.76	-23.9	2	11.85	-23.93	-47.87
x1	x5	0	14.42	15.57	0.064	-1.16	0.93	-7.4	1	14.42	-7.43	-7.43

H ROC curves for scATA algorithm applied to synthetic data

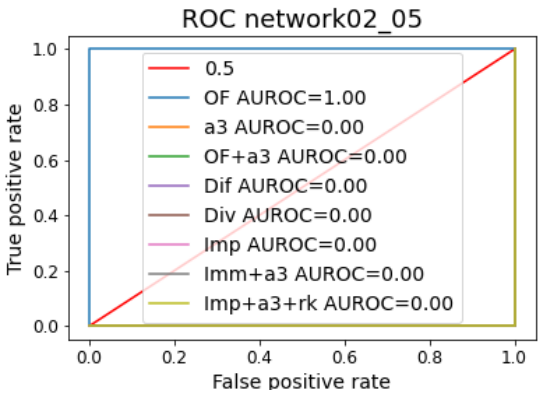
To evaluate the performance of the scATA algorithm, it was applied to the synthetic data generated in Section 2.3.2. The aim of the scATA algorithm is to infer the underlying GRN of the data. Therefore, a perfect performance of the algorithm would be to reconstruct the exact network topologies detailed in Appendix E. The following figures present the ROC curves for all the scores listed in 3.2.5. Additionally, the AUROC is detailed inside the figure for each of the scores evaluated.

GRNs with 2 genes

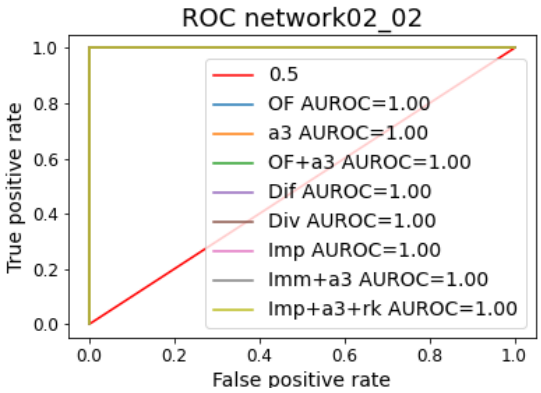
Network02_01



Network02_05

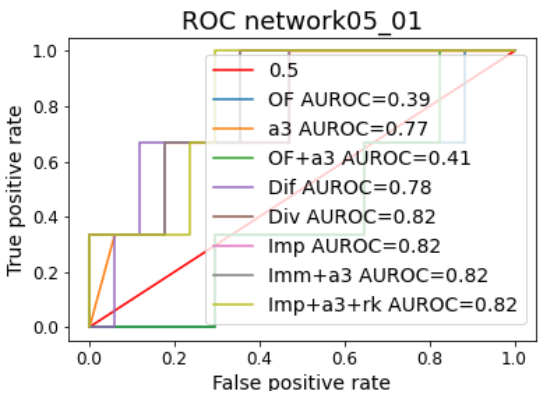


Network02_02

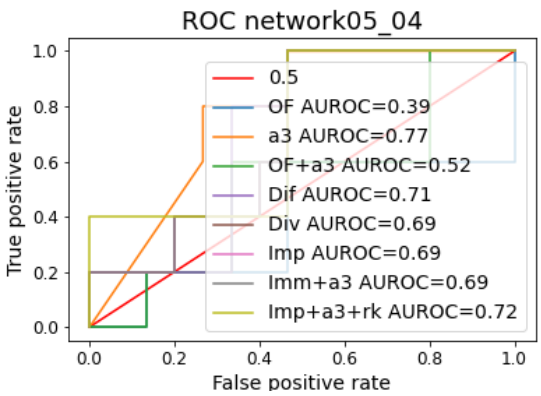


GRNs with 5 genes

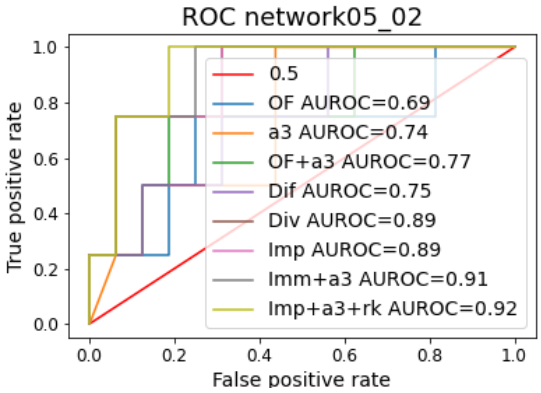
Network05_01



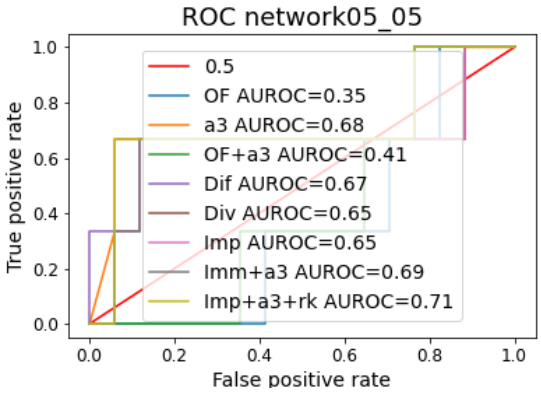
Network05_04



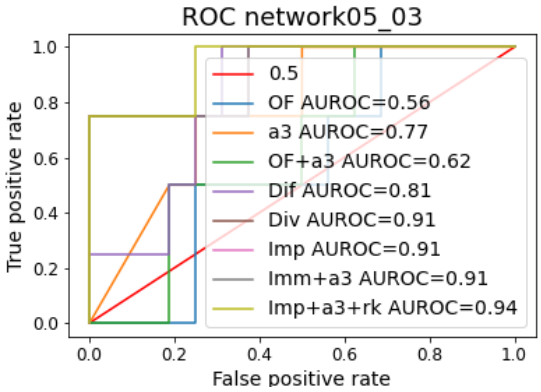
Network05_02



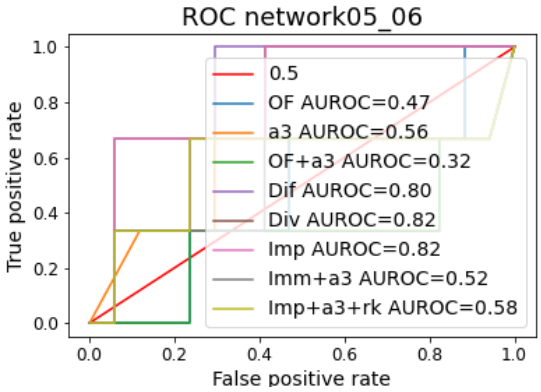
Network05_05



Network05_03

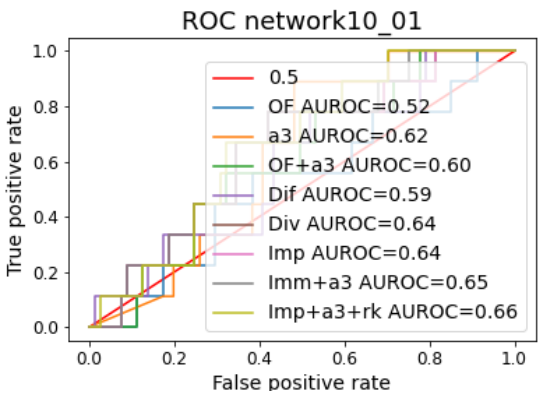


Network05_06

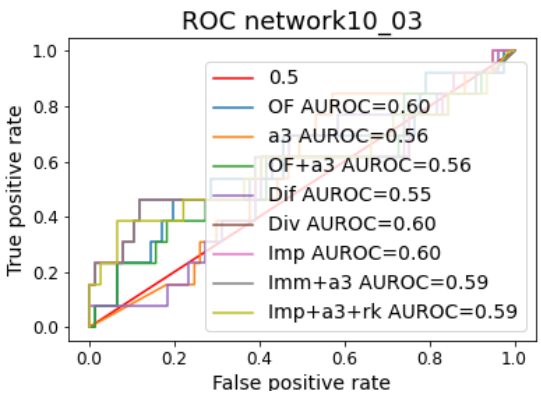


GRNs with 10 genes

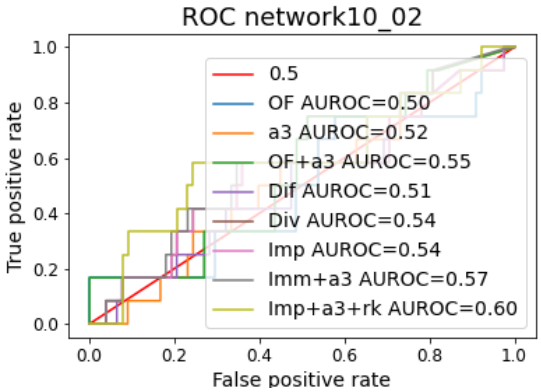
Network10_01



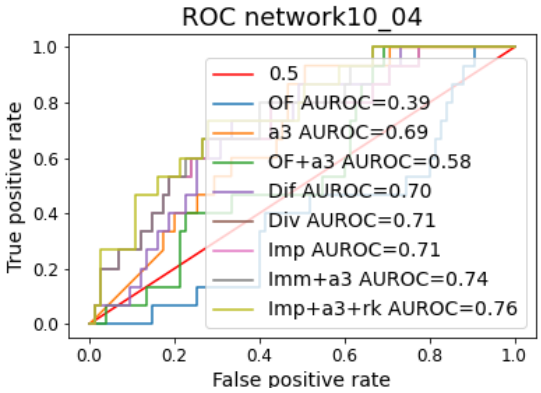
Network10_03



Network10_02



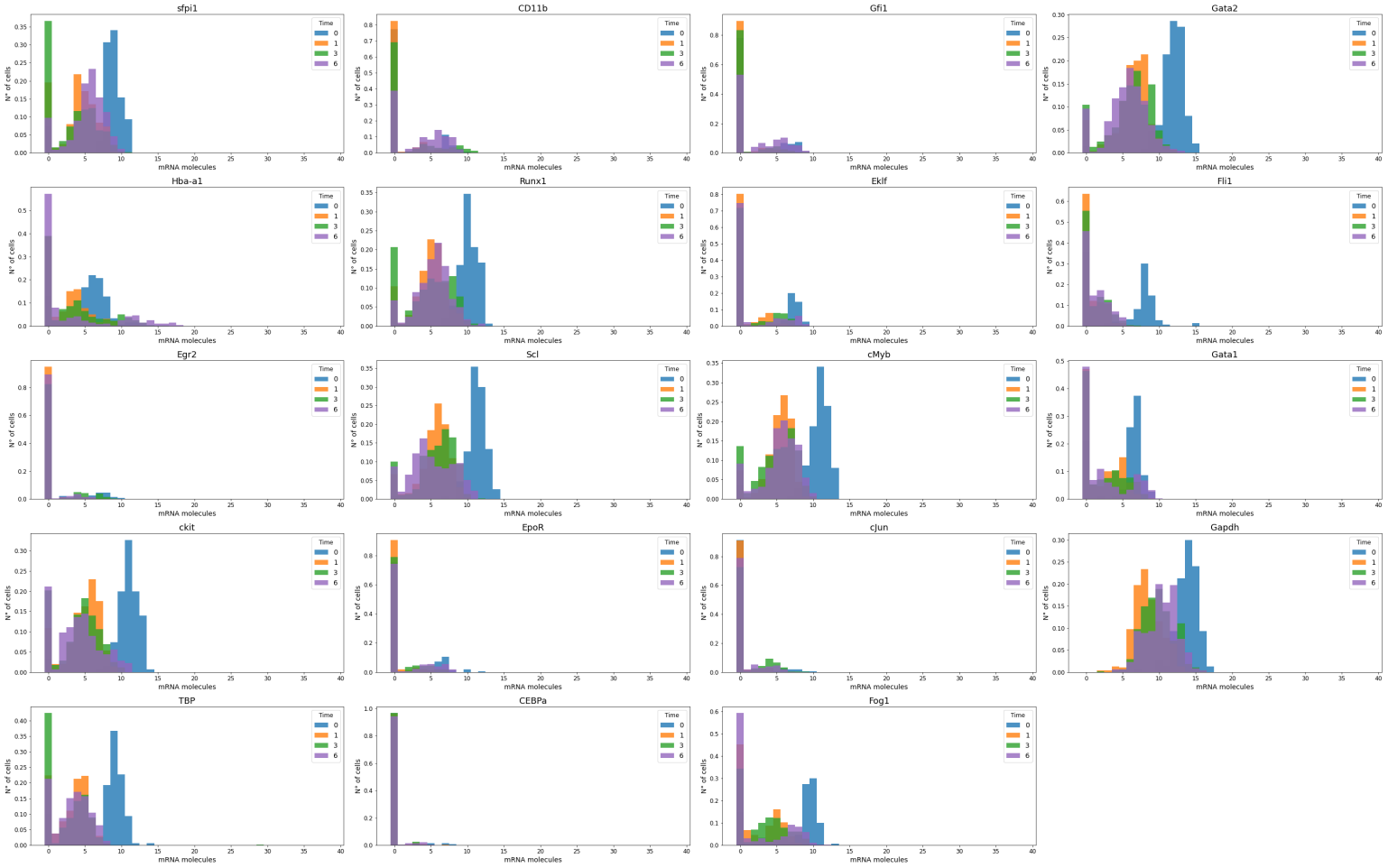
Network10_04



I Histograms of genes analyzed in the EML differentiation data set for the different treatments

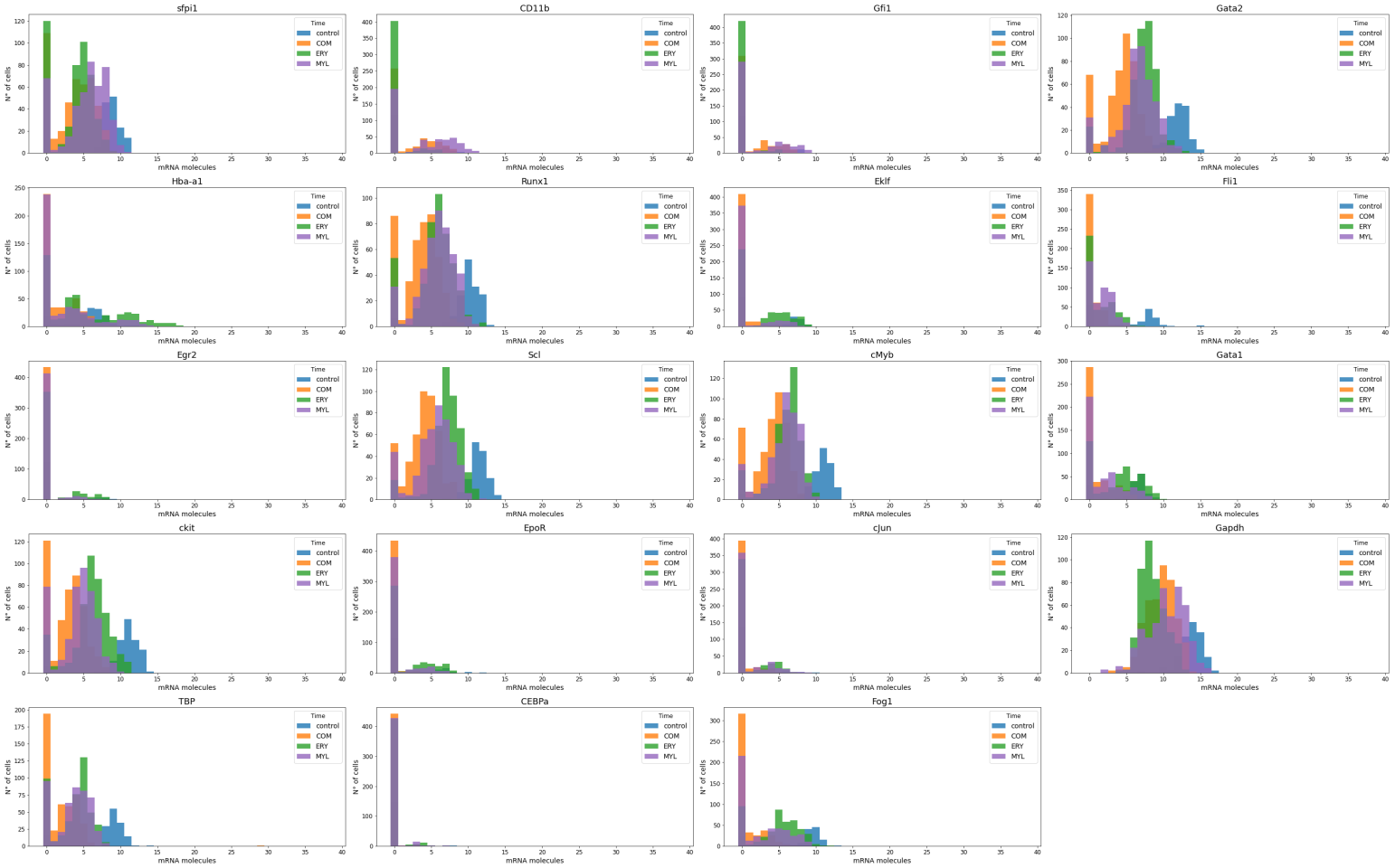
The different time points and treatments in the EML differentiation data set [43] were analyzed to see the trajectories of the genes present in the study. In this appendix, the histograms of the number of mRNA molecules for each of the different treatments (GM-CSF/IL-3, EPO, and GM-CSF/IL-3 and EPO), and all the cells together are presented.

ALL cells, different time points



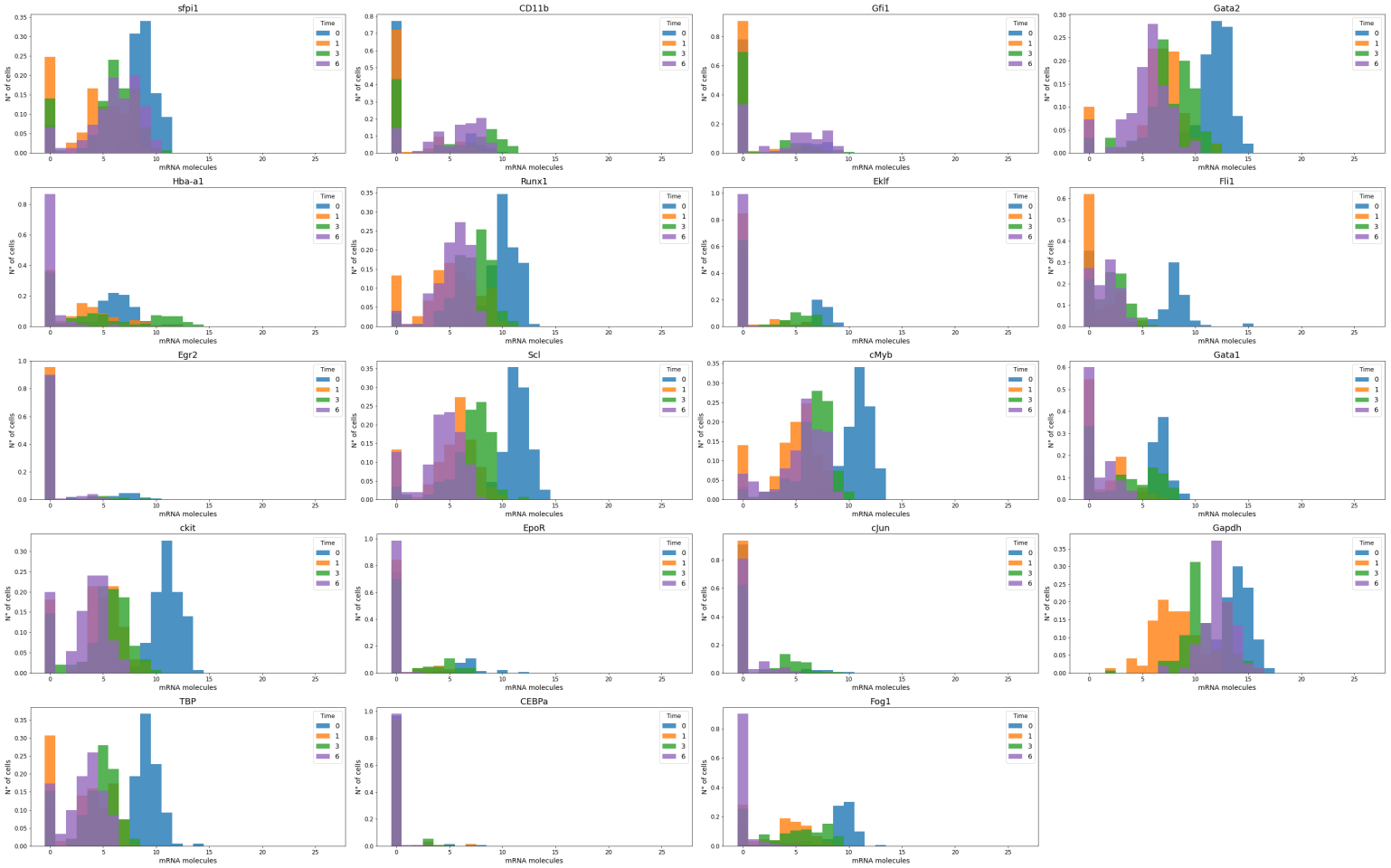
Snapshots of time points for the number of mRNA molecules for each gene for ALL cells in the data set.

ALL cells, different treatments



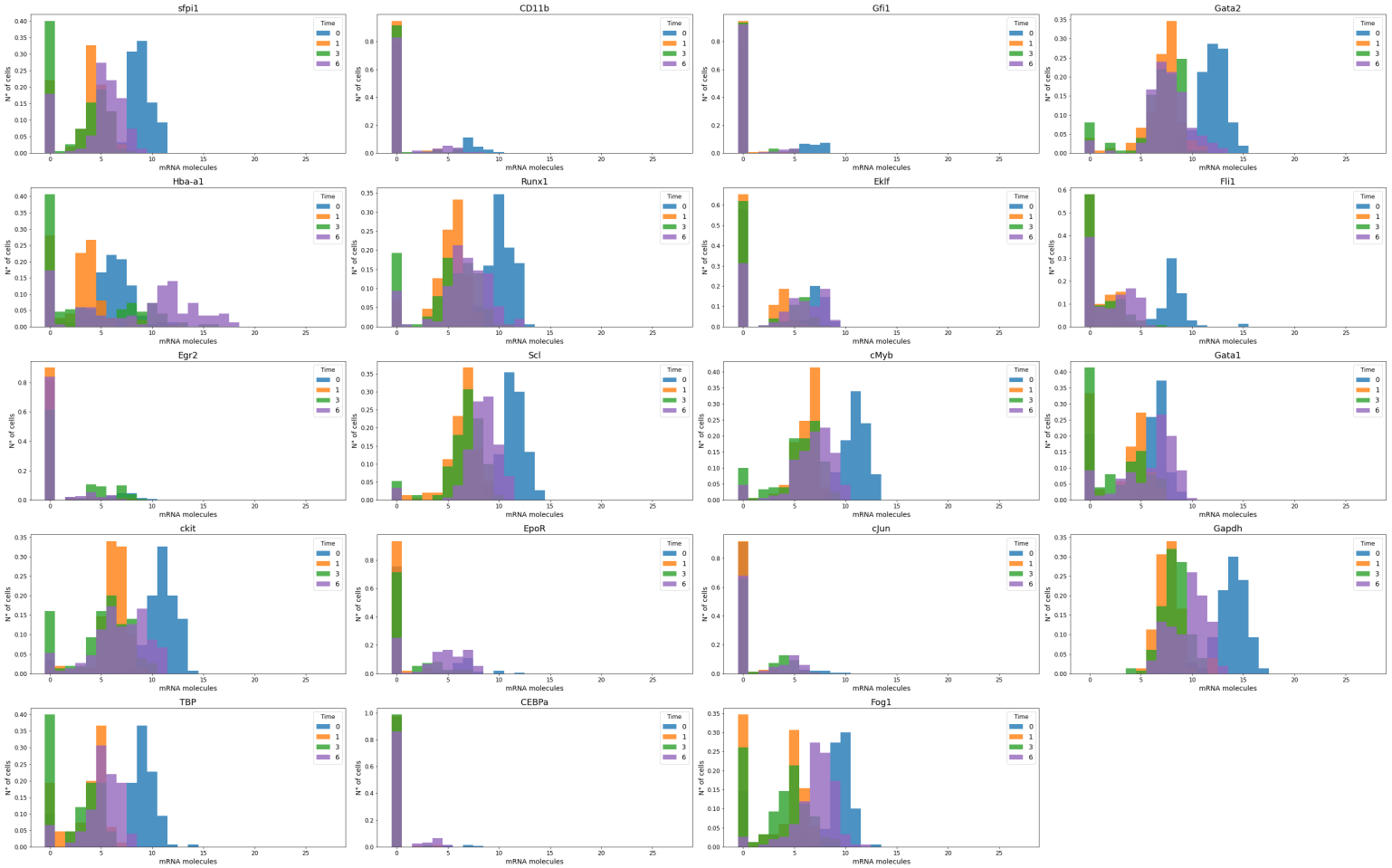
Histograms of the number of mRNA molecules for each gene for ALL cells in the data set, separated by treatment.

MYL cells



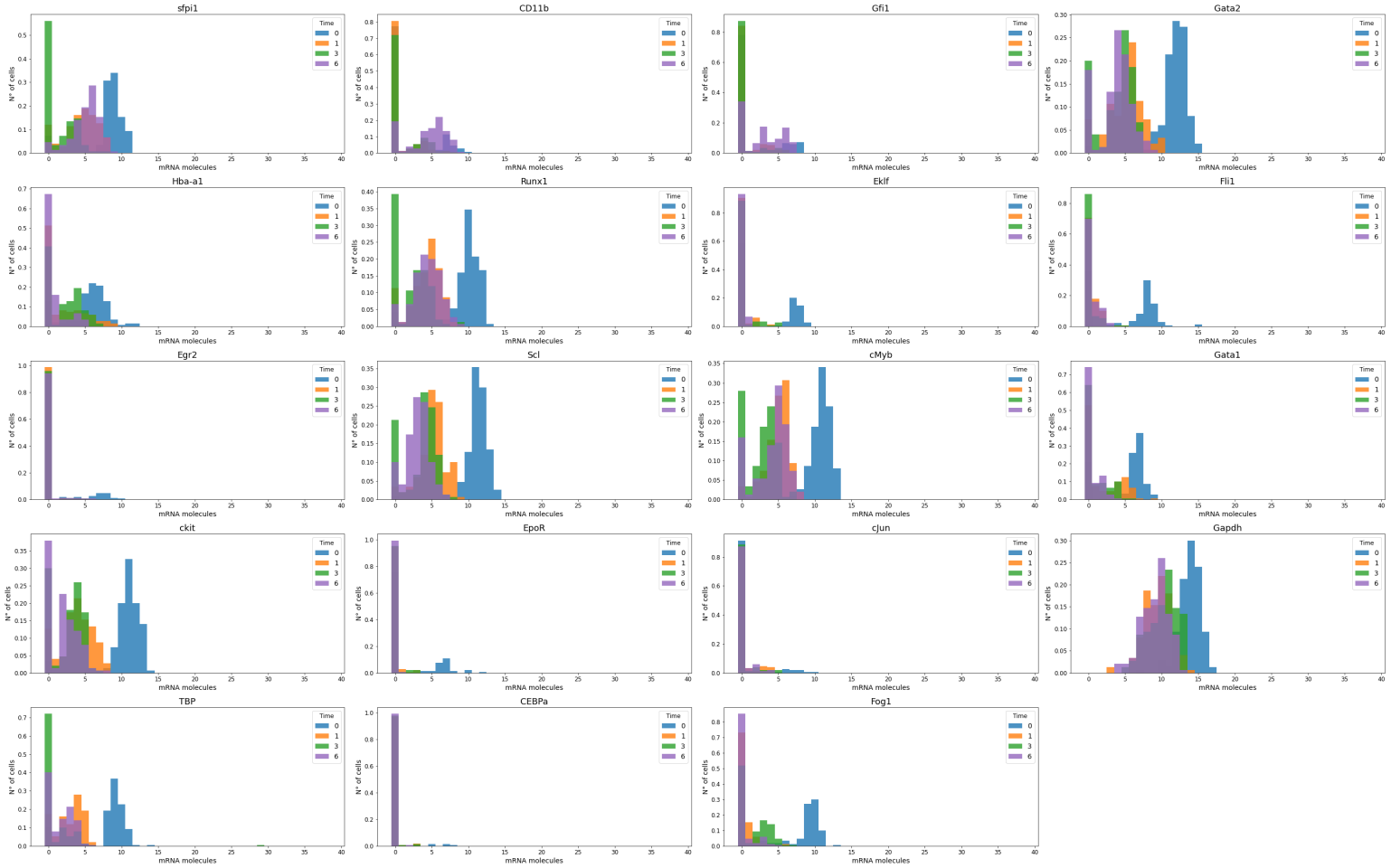
Snapshots of time points for the number of mRNA molecules for each gene for cells treated with GM-CSF-IL-3 (MYL).

ERY cells



Snapshots of time points for the number of mRNA molecules for each gene for cells treated with EPO (ERY).

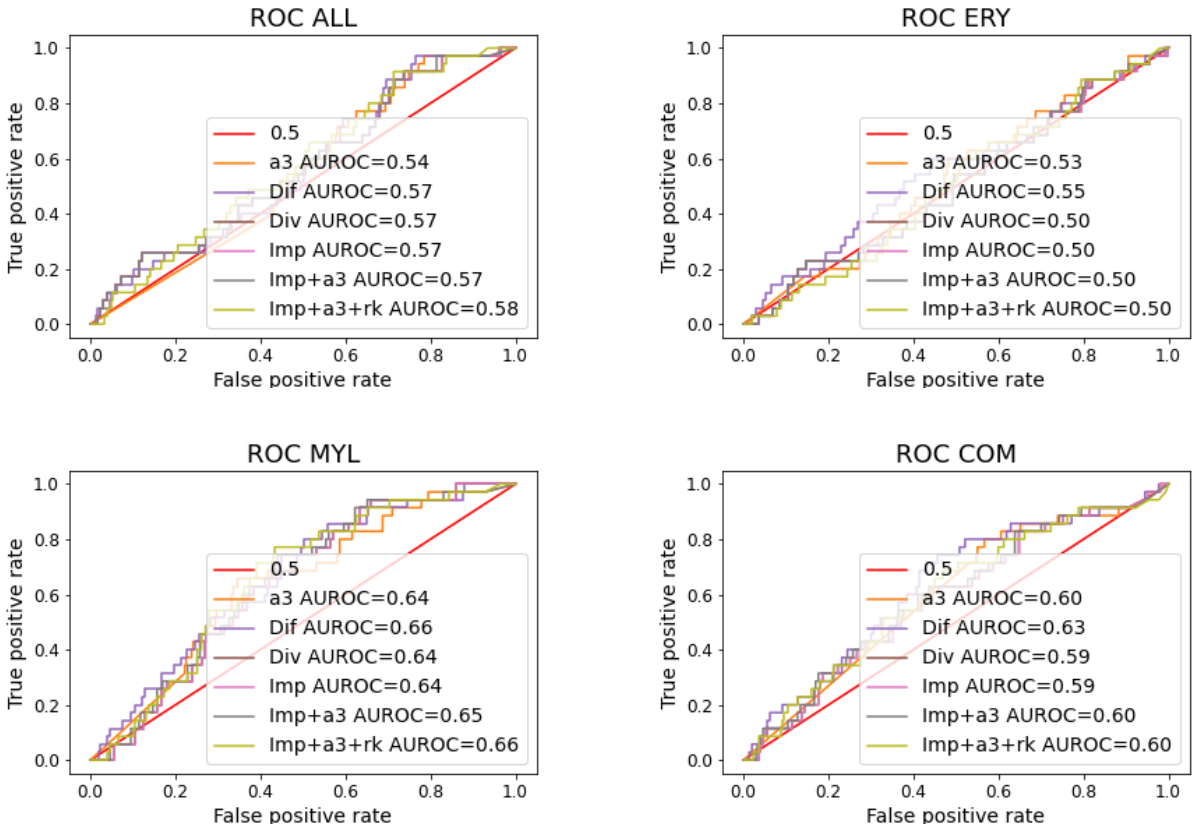
COM cells



Snapshots of time points for the number of mRNA molecules for each gene for cells treated with GM-CSF-IL-3 and EPO (COM).

J ROC curve of scATA algorithm applied to EML differentiation data set

The performance of our scATA algorithm was evaluated by using it to infer the underlying GRN from the EML differentiation data set [43]. The data set was separated by the different treatments of the cells (cells treated with GM-CSF-IL-3 (MYL), cells treated with EPO (ERY), and cells treated with GM-CSF-IL-3 and EPO (COM), and the algorithm used. The following figures present the ROC curves and AUROC for each of the treatments separately, and for all the cells together (ALL), for each of the scores evaluated.



ROC curve of scATA algorithm applied to EML differentiation data set.

Supplementary material

The python code used in the different chapters of this Thesis can be found in an online repository at:

https://github.com/MGRetamales/MISB-Thesis-MGR