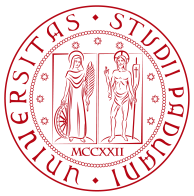


University of Padova

Department of Mathematics

Master Thesis in Data Science

Evaluation of RNA velocity as a signal for the reconstruction of gene regulatory networks

Supervisor

Professor Gabriele Sales

University of Padova

Master Candidate

Greta Farnea

Student number

2019052

Academic Year

2021-2022

To my beloved Nonna,
I hope
I made you proud.

Abstract

Single-cell sequencing techniques are becoming more and more used. Methods to compute RNA velocity are improving. In this work, the main purpose is to study in deep the relation between RNA velocity and the interaction of genes, in order to reconstruct the corresponding Gene Regulatory Network.

Contents

Listing of figures

1 Introduction

Single cell RNA sequencing (scRNA-seq) is an active research area in recent years which is rapidly developing and attracting attention due to its extraordinary potential. With this technology, it is able to observe the genome, the transcriptome and the proteomics at an individual cell level. It represents an evolution from bulk sequencing technique, whose data provides an average expression of a large population of cells. The scRNA-seq allows for the study of cell-to-cell variability as well and thus quickly generated enormous expectations for biologists and bioinformaticians. Indeed, it provides higher resolution data and a better understanding of biological processes of an individual cell within its microenvironment. It has become a standard technique to systematically discriminate cell types in mixed samples, to study cell differentiation and to better understand the dynamics of cancer cells.

Another fascinating possibility is to comprehend the interactions between genes and how they regulate each other. The observation of the change in gene expression among cells belonging to the same type enables to capture genes dependencies which can be described in a gene regulatory network (GRN). It should be taken into account that stochastic fluctuations and processes such as the cell cycle, which is always occurring in a living cell, are an important source of noise for this type of data. On account of this, scRNA-seq data brings new challenges for the mathematical modelling and the computational development of any methods.

The aim of this thesis is to study the already existing algorithm GRISLI [6] and to increase its performance by making use of more precise methods within its workflow. GRISLI estimates

the gene regulatory network starting from gene expression data, thanks first to the estimation of the RNA velocity and then to the solution of a sparse regression problem. The initial improvement consists in the employment of a preprocessing technique in order to remove some noise from the data. Then, starting from more reliable measurements, more advanced algorithms are investigated for the RNA velocity inference step.

The thesis is organised as follows. Chapter 2 introduces the biological processes of transcription and translation, fundamental for the comprehension of the definition of RNA velocity. In addition, scRNA-seq techniques are discussed in detail and the differences with respect to the bulk sequencing technologies are analysed. In Chapter 3 the concept of RNA velocity is defined and three algorithms for its inference are presented. Chapter 4 deals with the gene regulatory networks theory and it examines the GRISLI method in particular. Chapter 5 presents an overview of various techniques for the imputation of single cell data and for their simulation. Finally, in Chapter 6 the results obtained from the modification of GRISLI are shown, with some critical analysis about the limitation of this linear method.

2

scRNA-seq data

This chapter introduces the technology of single cell RNA sequencing data, its improvement with respect to bulk sequencing data and presents the phases of transcriptional dynamics. In Section 2.1, the fundamental concepts of transcription and translation processes are introduced. Section 2.2 explains the single cell technology pipeline, listing its strengths and also its limitations.

2.1 From DNA to proteins

Deoxyribonucleic acid (DNA) is composed by two chains of nucleotides, which are coiled around each other by hydrogen bonds, forming the characteristic double helix. The repeating units of the polymer are the nucleotides, which are only four: Adenine, that is complementary to Thymine and Guanine, whose complement is Cytosine.

The units of biological information is the gene, that is identified by a specific region of the DNA strand and which is responsible for a specific feature of the organism. In particular, genes encode for proteins that carry out necessary functions for the lives of the cell and of the whole organism. For example, some proteins must metabolise nutrients, others synthesise biological constituents or create copies of the DNA. In order to encode for proteins, transcription and translation processes must occur.

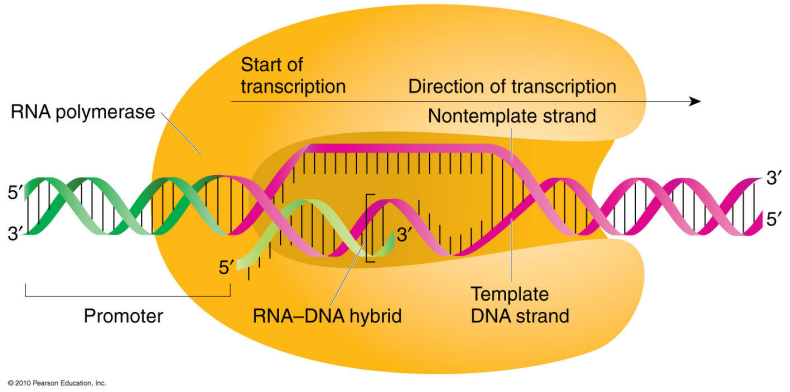


Figure 2.1: Schema of transcription process. RNA polymerase (yellow) separates the strands of DNA and the lower strand is taken as the template, for the synthesis the precursor mRNA (light green). The promoter region of DNA is highlighted in green.

During transcription, the genetic information is transferred to a messenger ribonucleic acid molecule (mRNA). It is really complicated since its setting up is regulated by cis-regulatory elements, such as promoters and enhancers, which are non-coding regions of DNA that are located respectively close to the gene or distant from it. Proteins called transcription factors bind to these non-coding regions and, thanks to many mediator proteins, are able to make the enhancers and promoters interact, in order to initiate transcription. At this stage, RNA polymerase, which is complex enzyme, binds to the promoter, which is still fully double-stranded. RNA polymerase can separate the strands of a small region of the DNA, such that one single strand can be used as a template.

When the enzyme finds the transcription start site, the elongation phase begins. Taking the single strand of DNA as a template, a chain of complementary nucleotides is build, thus forming a molecule of mRNA. It should be noticed that this molecule contains the same information of the coding region of the gene, where the Thymine nucleotide is replaced by the base Uracil. During the elongation, a process called capping occurs. It consists in the addition of an altered Guanine at the beginning of the transcript. This allows the precursor mRNA to be recognised as such and not be degraded. It also enables the correct occurrence of the splicing and facilitates the translation process.

The termination is characterised by a specific sequence which signals that the RNA transcript is complete. This process involves the cleavage of the new RNA transcript and the addition of few hundreds of Adenines, in a reaction called polyadenylation. This poly(A) tail is fundamental for the stability of the mRNA, protects it from degradation at that end and will influence the downstream translation.

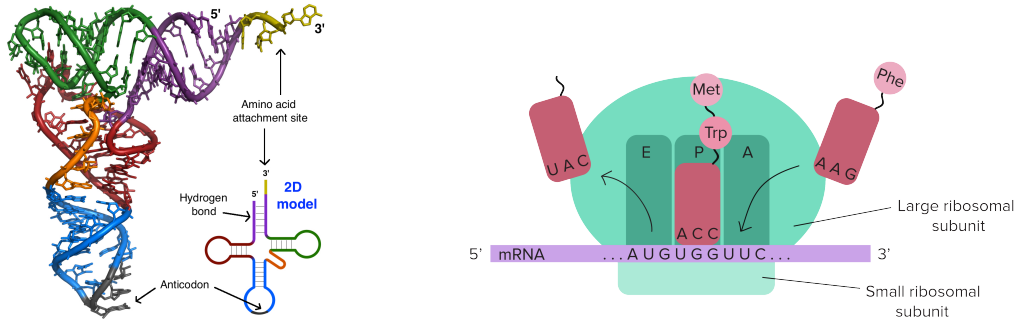


Figure 2.2: (a). The particular conformation of tRNA. The upper part is responsible for the transportation of the amino acid. The lower part is constituted by the anticodon, which will attach to the complementary codon on the mRNA. This tertiary structure is maintained thanks to hydrogen bonds. **(b)**. Schematic representation of translation. The mRNA sequence (purple) is bounded to the ribosomal sub-units (green). The large sub-unit is composed by three sites: acceptor (A), peptidyl (P) and exit (E). When a codon is exposed, a tRNA molecule enters in the A site and the peptide chain is elongated with the newly transported amino acid. The tRNA moves to P site and when a new codon is attached by another tRNA, it will exit by moving to the E site.

This newly transcribed mRNA is defined as premature mRNA and some extra processing must occur before translation takes place. In particular, pre-mRNA undergoes splicing process and is transformed into mature mRNA. The transcript contains regions called introns and exons. The first ones are non-coding regions of RNA that must be removed. The second ones are the actually coding sequences, that must be spliced back together, once the introns are cleaved by a complex called spliceosome. Therefore, the mature mRNA that is obtained is composed only by exons, as long as two untranslated regions at the terminations of the molecule (the cap and the poly(A)). Indeed, only the exons of a gene encode a protein and the removal of the introns prevents the generation of nonsense or pathological polypeptide chains.

The second major step in gene expression is translation, which exploits the mRNA molecule to build a sequence of amino acids. Before starting, it is necessary to say that the mRNA carries information for building the proteins and this information comes in codons, which are groups of three nucleotides. There are 64 codons (given by the combinations of the four nucleotides in triplets) and each of them correspond to an amino acid, according to the genetic code rules. The only exceptions are the three stop codons, which are not associated to any amino acid but have the role of triggering some reactions to terminate the translation at the end of the process. Another important triplet is the start codon, which encodes for the methionine and is fundamental for the initiation of translation.

In addition, two complexes are necessary for the protein synthesis. The first one is the transfer RNA (tRNA) which has a particular conformation and acts as an intermediary between nucleotides and amino acids. In particular, one site of the tRNA carries a particular amino acid molecule, that is associated with a specific nucleotide sequence (codon), whose complementary (anti-codon) is found at another site of tRNA. The other complex is the ribosome, which is a macro-molecular organelle and is constituted of two sub-units, the small and the large one. Three sub-regions can be identified in the ribosome: the acceptor (A) site, that allows one tRNA to enter; the peptidyl (P) site, which contains the polypeptide chain that is being built; and the exit (E) site, from which the deacylated tRNA molecule is released.

Like transcription, translation also occurs in three stages: initiation, elongation and termination. The small sub-unit of the ribosome assembles around a molecule of mature mRNA and the complex is joined by the large sub-unit. The mRNA molecule is read a codon at a time and, thanks to the tRNAs, the correct amino acid, which corresponds to the codon, is added to the polypeptide chain. As the mRNA is pulled through the ribosome, a tRNA molecule enters the acceptor site and binds the codon. The growing peptide chain, attached to the tRNA on P site, is linked to the amino acid carried by the new tRNA onto the A site. The mRNA shifts one codon-length, allowing the following codon to be exposed and read. In this way, the tRNA in site A moves to site P and the one on site P moves to site E and then it is released.

This process is repeated many times, as long as new codons are available and the chain is extended with new amino acids. The termination of translation happens when one of the three possible stop codon is met. This is recognised by the so-called release factors, that bind to the P site and separate the polypeptide chain from the tRNA. The newly synthesised protein is released, the two sub-units of the ribosome detach from each other and a new translation can start rapidly.

2.2 Single-cell data

Single-cell RNA sequencing is part of next-generation sequencing (NGS) technologies that are rapidly progressing in recent years. It provides valuable insights into the biological systems of individual cells, providing data with higher resolution than the previous technologies, such as bulk RNA sequencing, and enables a deeper understanding about the complexity of micro-environments. The objective of this technology is to generate omics data (genomics, transcriptomics and epigenomics) that allows a precise characterisation of cells profiles, going further

the traditional profiling methods, which analyse biological samples at a bulk-level. For example, rare cells such as cancer cells, are extremely heterogeneous within their population, and single-cell data can uncover unexpected gene-to-gene relationships, that are fundamental for the awareness about cancer evolution. In developmental studies, single-cell technology can reveal critical genes that trigger cell fate decisions and the trajectory of different lineages in development can be inferred. In addition, the characterisation of outlier cells, which are not detected by analysis of pooled cells, implicate a better understanding of drug resistance and cells in diseased states.

Conventional methods, like bulk RNA-seq, analyse a large population of cells at a time, providing only an average of genes expression. However, even if each cell of an organism shares identical genotypes, the transcriptome is determined by the expression of only a portion of genes. Moreover, evidence shows that the expressions may vary also within similar cell types, disproving the hypothesis that cells of a given tissue are homogeneous, as most methods assume. Finally, most biological samples contain mixed ensemble of cells and bulk RNA-seq data obscures important differences between cells of various types.

Single-cell RNA-seq technique constitutes an evolution with respect to bulk RNA-seq because the generated data is not an average of gene expression among all the cells, but aims to faithfully represent the real expression profiles of an individual cell. Even if this is an important turning point, some limitations are still present in scRNA-seq, both technical and biological. The stochastic expression and the low amount of starting material cause the data to be affected by high dropout and noise. Indeed, scRNA-seq technologies produce noisier and more variable data, which is challenging for computational methods to analyse. Nevertheless, experimental protocols and bioinformatics pipelines have become gold standard in the past few years and the various steps resemble those of bulk RNA-seq, except the first ones.

The very first step consists in the isolation of single cell from which transcriptome information is obtained. Many methods have been developed for this step, from the limiting dilution, which is not very efficient, to the encapsulation of individual cells in droplets, thanks to a microfluidic device. These techniques continuously improve in performances, decreasing the false positive rates and bias, and constantly decreasing their cost.

The second phase is the preparation of scRNA-seq library. The cells are lysed to allow the release of genomic and transcriptomic materials. Then, in order to capture the messenger RNA molecules, primers containing a segment of repeating deoxythymidines (dT) are adopted to an-

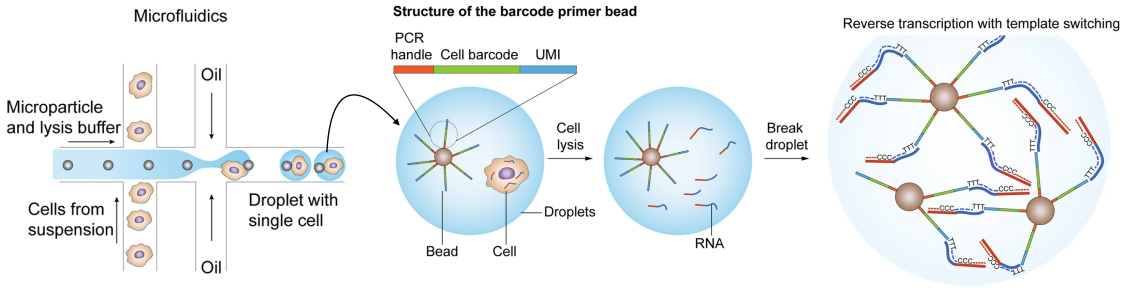


Figure 2.3: Image from [1]. Schema of microfluidic isolation approach. Cells are encapsulated into droplets, that contain primer beads identified by barcodes. Once the cell is lysed, fragments of RNA will attach to the beads and the reverse transcription starts. Then, the generation of scRNA-seq libraries is eventually completed with the cDNA amplification.

neal to the polyadenosine (polyA) tails present on mRNAs. At this stage, reverse transcription is initiated, yet only a small amount of those transcripts will complete this stage, around the $10 - 20\%$ of the total. The low efficiency of mRNA capture is one of the main limitations of this technology and require the adoption of a good upstream lysing approach. Molecules of complementary DNA (cDNA) are synthesised from the captured mRNA molecules, which serve as template, through a reaction catalysed by a chosen enzyme reverse transcriptase. This step is necessary for many reasons: DNA's structure is more stable, it allows amplification thanks to DNA polymerase and more developed DNA sequencing technology can be exploited. Since the amount of the obtained cDNA is relatively low, an amplification technique must be adopted. Conventional polymerase chain reaction (PCR) method can be used and one of its advantages is that permits to generate full-length cDNAs. However, PCR introduces biases since particular sequences may be amplified in an exponential manner, causing the creation of libraries with uneven coverage. Another approach for cDNA amplification is in vitro transcription (IVT), which avoid the coverage bias of PCR, but it can inefficiently transcribe specific sequences, causing dropout events and the generation of incomplete sequences. Since transcriptome analysis is performed on a huge amount of cells, barcodes of length 4 to 8 base pair or unique molecular identifiers (UMIs) are incorporated in the reverse transcription phase, so that each fragment of cDNA can be assigned to its original cell. It improves the accuracy and reduce the bias caused by the amplification step. It allows also for a better reproducibility than other read-based techniques.

Once the experimental laboratory procedure is completed, bioinformatics tools are necessary to analyse the amplified cDNA libraries. The first step consists in the quality control of the obtained reads, to get rid of low-quality bases and to remove the adapter sequences. Next, the

transcriptome must be assembled and two approaches are possible. One is the de novo assembly, which is used typically when a reference genome is not available. It consists in the generation of contigs, continuous sequences obtained by the union of several reads that could be adjacent in the genome. In order to find those reads, all-pairs overlaps graphs can be identified, but they are not efficient when dealing with millions of reads, or the de Bruijn graph approach can be exploited, which consists in the disruption of reads into sequences of k length (k-mers) that collapse into a hash table. Otherwise, if a reference genome exists, genome guided assembly is possible. This approach usually consists in the alignment of short portions of reads and then the use of dynamic programming, which results in the optimal alignment.

In general, only reads that map the genome with high mapping quality are retained for the construction of the gene expression matrix. The main feature of scRNA-seq expression matrix is the presence of zero-inflated value, caused by both technical limitations and transient gene expression. The gene expression is given by the number of reads that are mapped to the different loci in the transcriptome assembly phase. Then, normalisation of the data is commonly adopted to remove cell-specific bias. Scaling factors are used for this preprocessing step, and are obtained as the standardisation among cells. This procedure makes the strong assumption that most, if not all, genes are not differentially expressed. Other approaches are based on the between-sample normalisation, which assume that highly variable genes skew the abundance distribution in expression profiles. These methods are more efficient if the objective is to study the differential expression of genes.

It should be noted that all the downstream analyses based on scRNA-seq data have to deal with a high dimensionality problem. Indeed, the number of genes is usually bigger than the number of cells, which is an issue for computational methods. For this reason, it is common to adopt some reduction dimensionality approach, such as t-distributed stochastic neighbour embedding (t-SNE). It is a non linear method which is suited for the embedding of high dimensional data in a low dimensional space of two or three dimensions. This leads to an easy visualisation of the data and increases the interpretability of data and results. In particular, two probability distributions are defined over the pairs of data-points in the high- and the low-dimensional spaces. They are defined in such a way that similar ones are associated with a higher probability and different ones are assigned a lower score. Then, the method minimises the Kullback-Leibler (KL) divergence between the distributions according to a given similarity metric. Finally, clustering methods are useful for cell types identifications and can also be employed to detect low-quality cells, especially those enriched in mitochondrial genes.

After these preliminary steps, the processed data can be used for cell type characterisation, gene regulatory networks inference and trajectory inference. These studies are improved and more precise than when performed with bulk RNA-seq data, since data is more detailed and specific correlation between genes can be discovered, as well as less-expressed genes are not discarded, which usually happens when averaging the expression in bulk data.

3 Inference of RNA velocity

This chapter explains what is defined as RNA velocity, its possible mathematical representations and their limitation. In Section 3.1, it is introduced the biological notion of RNA velocity. Section 3.2 and Section 3.3 explore in detail two of the state-of-the-art methods for velocity inference. The last Section 3.4 discusses a different modelling approach, which however is at the core of this thesis.

3.1 The concept of RNA velocity

In the previous Chapter 2, it has been introduced the importance of single-cell transcriptomics. This kind of data can be exploited for the inference of RNA velocity and/or trajectory inference. These concepts enable the dynamical study of RNA processing from unspliced to spliced, through techniques described in proper *scRNA-seq* protocols. These procedures can distinguish between pre-RNA and mature RNA molecules, as a result of the presence of the so called *introns*. The latter are sub-sequences of RNA strands that do not directly code for proteins and which are removed during the splicing process.

Once stated the ability to discriminate between unspliced and spliced mRNA molecules, the theory of RNA-velocity can be introduced. In literature, in particular in [2], it is defined as the rate of change in mRNA abundance, which relates the unspliced and spliced mRNA quantities and it is induced by two main processes. The first one is the degradation of the mRNA

molecules, fundamental for the regulation of gene expression and for the elimination of defective RNAs. The second one is the splicing process, that generates spliced mRNA starting from unspliced one, and eventually permits the translation from RNAs to amino acid strands.

At the level of a single-cell measurement of RNA abundances, the velocity is a punctual temporal estimate, in the sense that no temporal relationships can be exploited. Moreover, it is a high-dimensional vector, whose dimension is given by the number of genes that are taken into consideration, and which could be used to infer the future state of the single cell. It must be recalled that each element of the velocity vector refers to the instantaneous velocity of one particular gene of the cell and its value describes its dynamics. Therefore, single values of velocity vectors can be interpreted according to their mathematical sign. Positive values indicate an instantaneous up-regulation of the specific correlated gene. This means that the abundance of unspliced mRNA is higher than expected and there is a deviation from the steady-state of the cell. Vice versa, negative values are related to down-regulation phases, during which the unspliced mRNA is less abundant. A null velocity means that the gene is in one of two steady-states, which have been hypothesised to occur. It may be repressed, meaning no unspliced nor spliced mRNA is observed, or it could be actively transcribed, while the rate of degradation and generation of spliced mRNA counterbalance each other.

This relatively straightforward concept relies on the strong and fundamental assumptions that the experimental data captures all the phases of a gene expression, from induction to repression. It is crucial since the RNA velocity is expressed relatively to the steady-state ratio of unspliced and spliced mRNA. The lack of this information could lead to an incorrect inference of RNA velocity and might compromise the drawn conclusions. As discussed in [12], this assumption is not always met because of many reasons. The experiment might not last enough to capture the whole biological reaction of interest, the up-regulation may start at the very end of the process or the down-regulation at the very beginning of it. Another problem could be that the difference between up- and down-regulation abundances is not detectable and, thus, the deviation from the steady-state ratio is not discernible.

For these reasons, even if the method implemented in [2] is widely used, some progress has been made [3], with the goal of removing some of those restrictions. Section 3.2 and Section 3.3 will describe in more detail this aspect.

Finally, the general formulation of the dynamic model of transcription can be introduced. Many of the state-of-the-art methods are based on two first order differential equations that

formalise three biological processes. Equations 3.1 are expressed with respect to one specific gene and can be easily extended to a multitude of them.

$$\frac{du(t)}{dt} = \alpha(t) - \beta(t)u(t)\,, \qquad \frac{ds(t)}{dt} = \beta(t)u(t) - \gamma(t)s(t) \tag{3.1}$$

In order not to lose generality, the variables and the parameters are all time-dependent. The variables $u(t) = (u_1(t), \ldots, u_n(t))$ and $s(t) = (s_1(t), \ldots, s_n(t))$ are multi-dimensional vectors, with n being the number of cells, and they represent the abundances of unspliced and spliced mRNA, respectively. The parameters describe the rates of the different processes: $\alpha(t)$ refers to the rate of transcription of precursor mRNA $u(t)$, $\beta(t)$ to the rate of splicing, that generates mature mRNA $s(t)$ and finally $\gamma(t)$ to the rate of mRNA degradation. On the left-hand side of Equations 3.1, the time derivative of both pre-mRNA and mature mRNA are defined, with only the latter ($\frac{ds(t)}{dt}$) being the RNA velocity.

This formulation is the most straightforward and general model that has been proposed and on which the literature is based nowadays. Certainly, it can be adapted according to the assumptions that different methods might make.

3.2 Steady-state model of velocyto method

After introducing the concept of RNA velocity, it must be said that different methods for its inference rely on slightly diverse dynamical models, whose attempt is to go around or totally avoid the previously listed problems. In particular, in the previous section, the explained widespread theory is formalised in *La Manno et al.* [2]. The paper introduces their algorithm *velocyto*, which is one of the first developed and state-of-the-art methods for the RNA velocity inference. This method is able to approximate the first time derivative of the gene expression state, by following a simple model for transcriptional dynamics, as shown in Equation 3.2 and in Figure 3.1b.

$$\frac{ds(t)}{dt} = \beta u(t) - \gamma s(t) \overset{(\beta=1)}{=} u(t) - \gamma s(t) \tag{3.2}$$

The parameters β and γ correspond to the splicing rate and the degradation rate and, differently from Equations 3.1, both of them are constant. Furthermore, it can be noticed that the parameter β is set to be equal to 1. It is a strong assumption, since it implies a constant and common splicing rate for all genes, that, as one might imagine, is not what happens in a real biological environment. Still, this simplification is useful in order to diminish the number of

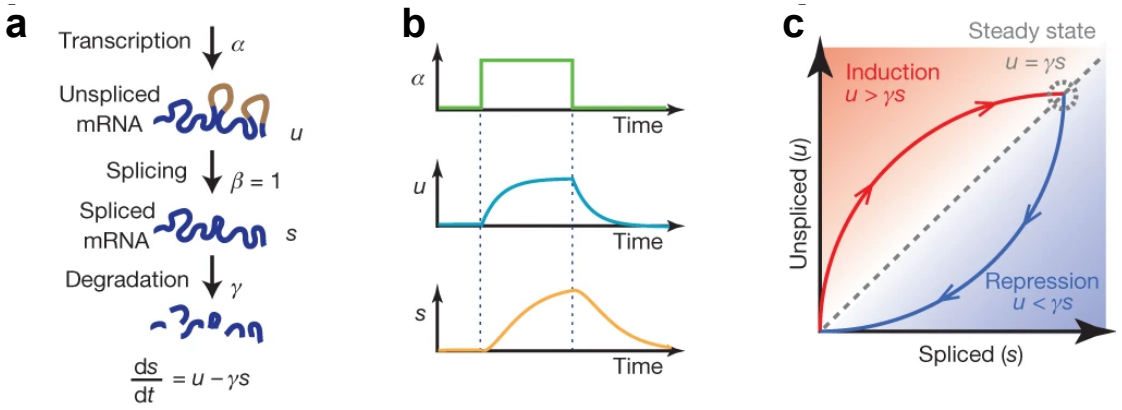


Figure 3.1: Image from [2]. **a.** Schema of transcriptional dynamics. Parameters represent transcription rate, splicing rate and degradation rate. At the bottom, the equation of velocity. **b.** According to a step change of α, it is represented how u and s dynamics react. **c.** Phase diagram: on x-axis there is s, on the y-axis u. Bottom-left equilibrium corresponds to passive steady state; top-right equilibrium to active steady state. Diagonal dashed line represents the slope γ of steady states for different values of α. Upper-space (red) represents positive velocities, i.e. up-regulation of a gene; lower-space (blue) represents negative velocities, i.e. down-regulation.

parameters to tune.

Recalling what said in Section 3.1, the strong assumption of this model is that both transcribing and silenced (i.e. active and passive) steady state equilibria are observed in the data. In those steady states, the synthesis of spliced mRNA and its degradation counterbalance each other and the velocity is null ($\frac{ds}{dt} = 0$). In order to properly tune the parameter γ, linear regression can be used on the lower and the upper quantiles in the phase space of mRNA expression, as they correspond to the silenced and active steady state respectively. As a result, a straight line can be drawn as in Figure 3.1c (grey dashed line) and it represents all the intermediate steady states where the ratio between spliced and unspliced amounts is constant and equal to γ. Once this ratio is found, induction and repression phases correspond to the upper- and lower-space with respect to the steady state diagonal. It is the reason why this sort of model is referred to as the *steady-state* model.

Analytically, the value of γ can be found through least square fit, by solving Equation 3.3.

$$\tilde{\gamma} = \beta \frac{u^T s}{||s||^2} \overset{(\beta=1)}{=} \frac{u^T s}{||s||^2} \tag{3.3}$$

Finally, the approximation of RNA velocity $\tilde{v}$ is computed as the deviation from the steady

state ratio, that is the just found $\tilde{\gamma}$ parameter (Equation 3.4).

$$\tilde{v}_i = u_i - \tilde{\gamma} s_i \quad \forall i \in \{1, \dots, n\} \tag{3.4}$$

A null velocity represents a constant transcription of that gene; whereas, a non-zero velocity indicates a dynamic process is happening. The direction and the rate of it depend on the mathematical sign and the absolute value of the velocity, respectively.

3.3 Dynamical model of scVelo method

The second state-of-the-art method for the RNA velocity inference is *scVelo*, which has been developed and presented by *Bergen et al.* [3]. It is a likelihood-based and dynamical model and allows the inference of specific reaction rates at the gene level. Moreover, it infers a *latent time*, which can be considered as a cell's internal clock, that places the cell at a point in an underlying biological process, and, as a result, is shared among the cell's genes.

The goal of this method is to overcome the restrictions introduced by the steady-state model. The first problem that has been noticed is that the passive and active steady states are not always observed in the data. The other restriction is the assumption that the splicing parameter β is common and constant among all the genes.

As shown in Figure 3.2b, a phenomenon called *early switch* is likely to happen, in particular among transient cell populations. The immature cells differentiate in mature cells, passing through temporary states which might be unstable. In [3], they consider various lineages in hippocampal dentate gyrus neurogenesis and pancreatic endocrinogenesis. The assumption of capturing both steady states discussed in Section 3.2 is inevitably violated and *scVelo* aims to manage also those cases. Indeed, the predicted latent time reconstructs with enough accuracy the temporal sequence of their transcriptomic events and the cellular fates.

The dynamics of this model is formulated in the following Equation 3.5.

$$\frac{du(t)}{dt} = \alpha^{(k)} - \beta u(t)\,, \qquad \frac{ds(t)}{dt} = \beta u(t) - \gamma s(t) \tag{3.5}$$

The parameters $(\alpha^{(k)}, \beta, \gamma)$ refer to the rates defined for Equation 3.1, but are now specific and different for each cell and for each time point $t \in \{1, \dots, N\}$.
To overcome the problem of non-observed steady states, the equations are explicitly solved by

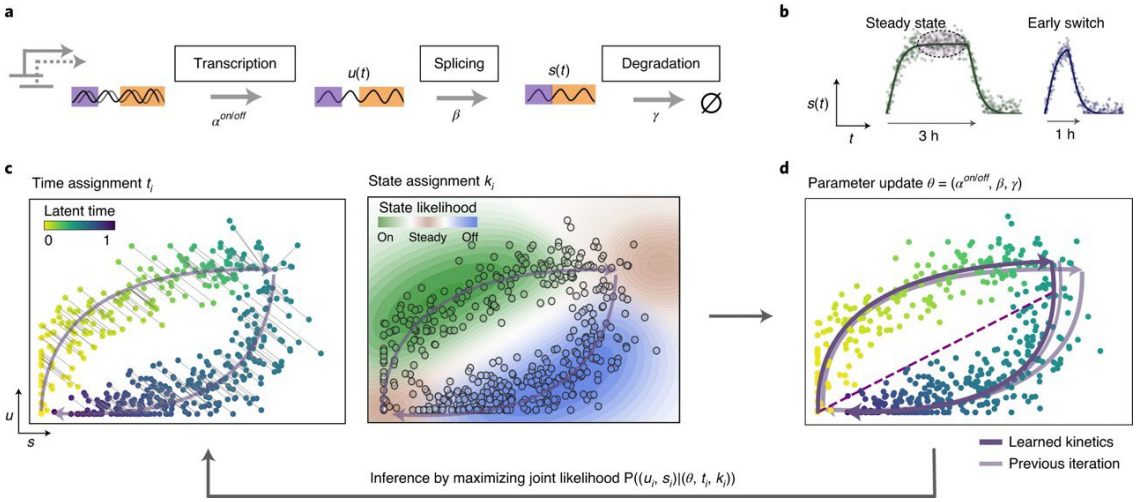


Figure 3.2: Image from [3]. **a.** Representation of transcriptional dynamics. Parameter α captures the induction and repression phases of pre-mRNA. Parameters β and γ account for splicing and degradation rates. **b.** The left plot describes a stable steady state where the transcription persists over time. The left plot shows a passive steady state, named *early switch*, in which induction terminates before unspliced mRNA saturation is reached. This particularly happens in transient cell populations. **c.** Two plots portray the transcriptional dynamics while highlighting the latent variables assignment. On the left, the latent time is assigned and projected onto the learned kinetics; on the right, sub-regions correspond to four internal states of the cell. A likelihood is associated to those variables and will be needed to update the other parameters. **d.** After the latent variables have been fixed, transcriptional parameters are updated and so it is the gene-dynamics. Finally, the successive iteration starts by going back to the assignment of latent variables and their likelihood.

integration and the solution is found in Equation 3.6.

$$\begin{aligned} u(t) &= u_0\, e^{-\beta\tau} + \frac{\alpha^{(k)}}{\beta}(1 - e^{-\beta\tau}) \\ s(t) &= s_0\, e^{-\gamma\tau} + \frac{\alpha^{(k)}}{\gamma}(1 - e^{-\gamma\tau}) + \frac{\alpha^{(k)} - \beta u_0}{\gamma - \beta}(e^{-\gamma\tau} - e^{-\beta\tau}), \quad \tau = t - t_0^{(k)} \end{aligned} \tag{3.6}$$

The initial condition for unspliced and spliced mRNA are given by $u_0 = u(t_0)$ and $s_0 = s(t_0)$. The kinetics is described by two sets of parameters: the first one is given by $(\alpha^{(k)}, \beta, \gamma)$ and the second one includes cell-specific latent variables, which are a transcriptional state k and a continuous time $t \in [0, 1]$. In particular, k is a discrete variable that takes values in {on, off, ss_{on}, ss_{off}}. These labels respectively refer to the induction and repression phases, or to the active and passive steady states.

The two sets are interdependent and their estimate can be obtained through the expectation-maximisation process. In the first step, given an approximated phase trajectory $\mathcal{X} = (\hat{u}(t), \hat{s}(t))_t$, a latent time t_i is assigned to each mRNA pair $x_i = (u_i, s_i)$, according to the minimization

of its distance to the learned phase trajectory χ (Figure 3.2c). Meanwhile, it is also assigned a transcriptional state k_i to each mRNA abundance x_i, by associating a likelihood to different portions of the trajectory χ. In the second step, the likelihood is maximised through the update of the transcriptional rates (Figure 3.2d). Finally, the method iterates those two steps until it reaches the convergence for genes showing a clear kinetics.

Once the expectation-maximisation process has finished, RNA velocity is predicted as the derivative of mature mRNA abundance, that is characterised by the explicit description of the previously optimised splicing kinetics.

3.4 Three-point model

In the preceding sections, two state-of-the-art models have been examined. Their mathematical foundation is Equation 3.1, which models the dynamics of unspliced and spliced mRNA abundances as a system of two linear differential equations. Their inference is based only on the measurements deriving from scRNA-seq experiments.

However, the starting point work of this thesis is [6], in which they use a different concept and computation for the RNA velocity. Their objective is not the velocity inference, but rather it is the gene regulatory network prediction (Chapter 4). Therefore, their approximation of the velocity is much more basic and it is based on the rate of change in the mature mRNA molecules and the *pseudo-time* label.

The model is completely different from Equation 3.1 and none of the parameters which have been encountered before is used. It describes the expression process through a linear ordinary differential equation, formalised in Equation 3.7.

$$\frac{dx(t)}{dt} = \boldsymbol{A}x(t) \tag{3.7}$$

The variable $x(t)$ is a multi-dimensional vector representing the gene expression (i.e. the mature mRNA abundance) for each cell. The pseudo-time label t is assigned to each cell and it is a numerical description of where the cell is in the transcriptional process. $\boldsymbol{A}$ is a square binary matrix that characterises the dependencies among the genes. In particular, its dimension is defined by the number of genes that are considered. Each column and each row corresponds to one gene and an element of the matrix $a_{ij} \in \boldsymbol{A}$ is non-null if and only if there exists a regulatory interaction between the i-th and the j-th genes. It is of crucial importance, since it captures the internal relationships between genes and it is considered to be one of the main

reason behind the change of spliced RNA abundance.

This section is concentrated on the left-hand side of the equation, while the next Chapter 4 will discuss the whole structure in detail.

The RNA velocity inference relies on the pairs (x_i, t_i) of spliced RNA measure and time-label for each cell. A first estimation is done with respect to any other cell (described by (x_j, t_j)) with a different time-label, through the finite difference expressed in Equation 3.8.

$$\hat{v}_{i,j} = \frac{x_i - x_j}{t_i - t_j}\,, \quad t_i \neq t_j \tag{3.8}$$

It can be noticed that if t_i and t_j are too far, the finite difference is no more a good approximation of the derivative, since the delta-time should be as small as possible. The same reasoning applies to x_i and x_j: if two cells' trajectories are too distant, their difference becomes meaningless.

The real estimate of the velocity $\hat{v}_i$ of cell i is given by the weighted average of all the $\hat{v}_{i,j}$, with $j \neq i$. The weights in Equation 3.9 are defined as a spatio-temporal function $K(x, t, x', t')$ that evaluates how much (x', t') is meaningful in the velocity inference at (x, t) point.

$$\hat{v}_i = \frac{1}{2} \underbrace{\frac{\sum_{j|t_j>t_i} K(x_i, t_i, x_j, t_j)\, \hat{v}_{i,j}}{\sum_{j|t_j>t_i} K(x_i, t_i, x_j, t_j)}}_{\text{future}} + \frac{1}{2} \underbrace{\frac{\sum_{j|t_j<t_i} K(x_i, t_i, x_j, t_j)\, \hat{v}_{i,j}}{\sum_{j|t_j<t_i} K(x_i, t_i, x_j, t_j)}}_{\text{past}} \tag{3.9}$$

It can be noticed that Equation 3.9 is composed by two weighted averages: the first one regards points in the future with respect to the considered t_i, while the second one concerns the points in the past. This is needed because the relative position in time acts differently on the estimation of $\hat{v}_i$.

For the sake of completeness, the kernel function $K(\cdot)$ is defined arbitrarily by Equation 3.10, where σ_t and σ_x are constants and computed as the square root of the 10th percentile of the distribution of squared distances in time, t, and gene expression, x, respectively [6].

$$K(x, t, x', t') = (t - t')^2 \exp\left(-\frac{(t - t')^2}{2\sigma_t^2}\right) \times \exp\left(-\frac{\|x - x'\|^2_{\mathbb{R}^G}}{2\sigma_x^2}\right) \tag{3.10}$$

4

GRN reconstruction

This chapter addresses the topic of gene regulatory networks and some of the existing methods for their inference. In Section 4.1, the concept of gene regulation is explored from a biological and mathematical point of view. Section 4.2 introduces some inference methods, their assumptions and limitations in terms of performance. In the last Section 4.3, the second step of GRISLI regarding the regulatory network inference is illustrated in detail and a discussion about its performance is addressed.

4.1 Gene regulatory networks

A Gene Regulatory Network (GRN) is a high level abstraction of the transcriptional mechanisms, whose aim is to describe how genes and other regulators interact with each other. The regulator molecules could be DNA, RNA, proteins or more complex structures. One particular class of proteins, the *transcription factors* (TFs), exists only to activate or deactivate transcription processes of specific genes. GRNs usually comprehend mostly this class of proteins, at the expense of the *target genes*, which have other functions rather than regulatory ones.

The GRN can be represented as a proper network (Figure 4.1), composed of nodes and edges, which could be directed or not. Nodes denote genes, proteins or any other biological molecules that interact and control the gene expression levels. Edges express the existing relation between two nodes, that could be of two types: inductive, where the objective of one gene is to activate the expression of the other one; or inhibitory, in which case the objective is to

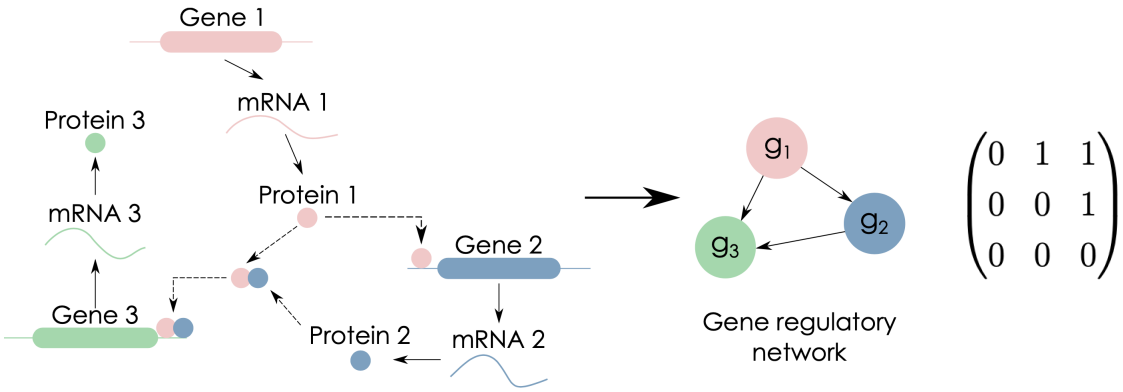


Figure 4.1: Image from [4]. The schema represents how a complex model (left), that includes three level of expressions (i.e. DNA, RNA and proteins), can be reduced by retaining only the interactions between the genes (right). On the most right side, the corresponding adjacency matrix is shown.

diminish or terminate the expression of the other gene. However, even if the most complete representation contains nodes for genes , mRNA and proteins, it must to be considered that one gene encodes for one mRNA strand, which will be translated into an amino acids chain. There exists a sort of a bijective relation between all these transcriptional products and therefore, the network can be simplified a lot, denoting all the interactions with respect to the genes only. From such a complex structure, as depicted in the left part of Figure 4.1, only the gene-wise relationships can be retained, in order to obtain a more schematic model.

From a mathematical point of view, such a network can be defined as a graph.

Definition (Graph). A graph is a pair $G = (V, E)$, where V is a finite set of vertices and E is a set of paired vertices, called edges. An *undirected* graph has edges connecting nodes symmetrically; a *directed* graph has nodes connected a-symmetrically, in the sense that if edge (i, j) exists, the inverse edge (j, i) does not necessarily exist.

Talking specifically about GRNs, different meanings can be assigned to the edges of the networks. They could describe qualitatively the interaction between genes, if it is inhibitory or inductive only, or they might be associated with the actual strength of the dependency, in which case the network is defined as *weighted*. Generally, in the context of GRNs, their associated value belongs to the set $\{+1, 0, -1\}$, with the significance of activating, null or inhibitory effect of one gene onto another one.

It is usually interesting to study some mathematical properties of the network, which acquire relevant meanings biologically speaking. For example, the *degree* of a node is defined as

the number of edges that are incident to the node itself. The higher the degree, the more crucial the gene function is within the regulatory network, making the gene to be considered as a *hub*. In addition, for directed networks, one could distinguish the *in-degree* from the *out-degree*. The first term refers to the number of arches terminating at the node, meaning how many TFs are regulating the gene, while the second one to the edges outgoing the node, standing for how many target genes it is responsible to regulate. This distinction is fundamental to discriminate which gene is responsible for the regulation of the other one, outlining the causal relation between the two of them and identifying which acts as the transcription factor and which as the target gene.

The last mathematical aspect of graph theory that is possible to exploit, is the possibility to construct a matrix, starting from the topology of the regulatory interactions, identified by nodes and the edges.

Definition (Adjacency matrix). An adjacency matrix is a square matrix $\boldsymbol{A} \in \mathbb{R}^n$ that describes a graph G of order n, in which each row and column corresponds to a node in the graph G. Each element $a_{ij} \in \boldsymbol{A}$ takes a value according to the fact that vertex i is adjacent to j: it is 1 if the edge (i, j) exists, otherwise it is 0. In the case of an undirected graph, the matrix is symmetric with respect to the main diagonal.

It can be observed that the matrix for GRNs is usually very sparse, i.e. presenting a lot of null entries. This characteristic is often problematic when it comes to modelling and inferring the matrix with the usual computational methods, as it will be discussed in later Sections.

4.2 Overview of GRNs inference methods

The single cell technology continues to rapidly develop, leading to the possibility of better understanding the cellular pathways. As discussed in Chapter 2, single cell data is characterised by features that are not present in bulk data. Thus, its statistical and bioinformatics analysis is more complicated and requires additional attention. For this reason, bulk-data based methods for GRN inference have a high probability of performing poorly and consequently providing ambiguous results, if applied on sc-data. One fundamental complication is the high percentage of zero values in single cell measurements, that may really challenge the performance of already-existing inference methods. This problem cannot yet be avoided because the presence of many zeros in the data is caused by two main reasons. Firstly, it is not expected that all transcriptional processes are happening in a single cell, and secondly, technical limitations are due

to the insufficient amounts of mRNA molecules which can be detected. For bulk samples, a standard preprocessing step is to impute the zero values in order to stabilise the inference methods. This technique applied on sc-data is still to be refined, since a possibility exists to alter the overall distribution of gene expression.

These reasons brought the scientific community to develop new GRN inference methods specifically for single cell experiments, whose goal is to extensively exploit the features of sc-data. It must be taken into account that this is a relatively new research field and so, these models are extremely simplified and the predicted network may not describe the entire regulatory mechanism of single cells.

In this section, the following notation for the GRN inference modelling is used. The number of gene present in the network is n and the number of sampled cells is C. The matrix X representing the gene expression has dimension $C \times n$, where the rows are n-dimensional vectors of transcriptome, and the columns are C-dimensional vectors, representing the gene profile among the cell population. The GRN inference methods start from the gene expression matrix X and predict the network of interactions between pairs of genes. The usual output is the adjacency matrix, where a non-null entry implies a connection between the two associated genes that could be caused by physical interaction or by an indirect one.

A series of benchmarking works have compared the performances of many GRN inference methods applied on sc-data, even the methods that were developed for bulk experimental samples ([5], [13]).

The most straightforward analysis that could be performed on sc-data is a correlation investigation. If two genes have very similar profiles, there is a high chance that they are interacting directly or are involved in the same regulatory process. This condition is however not sufficient, because the correlation may depend also by the effect of other variables. The measure of *partial correlation* can be computed as in Equation 4.1, where X_i and X_j are the i-th and j-th genes' expression vectors, S_m is the set containing all the other nodes and the variable σ_{ij} is the covariance between i and j.

$$\rho_{ij|S_m} = \text{corr}_{X_i X_j|S_m} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m}\,\sigma_{jj|S_m}}} \tag{4.1}$$

This method infers the non-zero values for $\rho_{ij|S_m}$ that indicate the presence of an edge between nodes i and j.

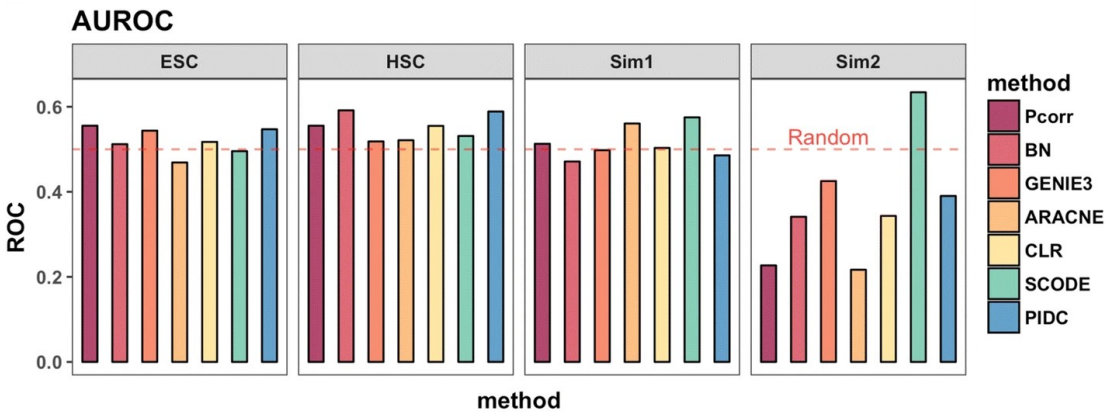


Figure 4.2: Image from [5]. The graphs shows the performance of different GRN inference methods (list at the right) applied on four datasets. ESC and HSC are real sc-data obtained from works which studied embrionic stem cell and blood-forming stem cell population respectively; while Sim1 and Sim2 are datasets simulated using GeneNetWeaver software (GNW). The dashed horizontal line represents the threshold of 0.5 corresponding to the random baseline for AUROC. It is clear that no method can consistently perform better than the random guess. Moreover, no method seems to stand out from the others.

A state-of-the-art method that is taken into consideration for the benchmarks is SCODE [14]. It infers the GRN topology of a single cell experimental data exploiting an ordinary differential equation that describes the regulatory dynamics: $dX = \boldsymbol{A}X\,dt$. The matrix $\boldsymbol{A}$ is the square adjacency matrix representing the gene interactions and X is the matrix of gene expressions. dX and dt represent the infinitesimal change in the expression data and in the temporal dimension respectively. For this reason, the method requires the pseudo-time data as an additional input. It is usually computed by external methods, such as Monocle [15].

Besides these two methods, there exist many others that are based on different assumptions. For example, Bayesian Networks or tree-based methods (GENIE3 [16]), methods like ARACNE [17] that use Mutual Information at their core.

Regardless of the theory which one method is based on, the conclusion drawn by these benchmarks is that the performances of all known methods are poor, from many points of view. In [5], depending on the method that is considered, the maximum value of Area Under Curve (AUROC) fluctuates around 0.5, that is the baseline obtained by random generated matrices (Figure 4.2). To the authors' surprise, the methods are more challenged by the simulated datasets, as the AUROC values are lower than the ones obtained for the real sc-data. This is probably due to the fact that simulators are not yet able to reproduce the characteristics of single cell measurements and those data might be a further challenge for the GRN inference

methods. Moreover, almost all the methods examined in [5] have high rates of false positives, that is defined as the percentage of predicted edge that is not present in the ground-truth network.

To recapitulate, these studies show that the GRN inference is still an open problem and many challenges still need to be overcome. This kind of method is a computationally demanding task and no single solution has been proven to exist. Furthermore, when it comes to evaluate the performances and the predicted networks, it must be kept in mind that regulatory networks are not yet understood comprehensively: therefore, when comparing the predicted network with the ground-truth GRN taken from the literature, the latter can be inaccurate itself. For this reason, any evaluation is always inherently incomplete and the networks used as ground-truth are not really representative of all the gene interactions taking place in a single cell.

4.3 GRISLI method

In Section 3.4 it has been introduced the work [6], that is the main core of this thesis, and it has been discussed the velocity inference method based on the three-point model. Now, the discussion of GRISLI can be concluded by examining the GRN inference part of the method.

It must be recalled the linear ordinary differential equation that GRISLI is based on and that describes the dynamics of cell expression with respect to a small temporal variation (Equation 4.2).

$$\frac{dx(t)}{dt} = \boldsymbol{A}x(t)\,. \tag{4.2}$$

The transcriptomic profiles $x(t) \in \mathbb{R}^G$ represent the vector of abundances for each one of the G genes. The pseudo-time variable t is a vector belonging to the set $\mathbb{R}^C$ and its entries are temporal labels for each cell. It can be assigned according to either the real experimental time or calculated by an external software (such as Monocle [15]). Consequently, the data given as input is a set of paired expression vector and time label: $\{(x_i, t_i) \in \mathbb{R}^G \times \mathbb{R} : i = 1, \dots, C\}$. Finally, the matrix $\boldsymbol{A} \in \mathbb{R}^{G\times G}$ is the matrix of GRN that has to be inferred. In particular, $a_{ij} \neq 0$ means that the j-th gene regulates the i-th gene. The matrix is assumed to be sparse, since the authors believe that each gene is regulated by only a few transcription factors.

GRISLI is a two-step method that firstly computes the RNA velocity of each cell $\hat{v}_i$, which is an estimate of $v_i = dx_i/dt$ (Section 3.4), and then it infers the regulatory interactions by estimating non-zero elements of $\boldsymbol{A}$ by solving a regression problem.

Once the approximation of RNA velocity has been accomplished, the second step consists in the GRN inference. The Equation 4.2 is considered as a sparse regression problem of the form $\hat{v} = \boldsymbol{A}x$. A score $s(i,j) \in (0,1)$ is computed for each pair of genes (i,j) belonging to $\{(i,j) : i,j \in \{1,\dots,G\}\}$, where j is believed to be the regulator and i the target gene. The score increases when it is believed that the entry a_{ij} is non-null, meaning that gene j regulates gene i.

The procedure through which $s(\cdot,\cdot)$ is computed, is proposed by [18] and involves three hyperparameters: $R, L \in \mathbb{N}$ and $\alpha \in [0,1]$. The expression data of all the cells is denoted as the matrix $\boldsymbol{X} = (x_1,\dots,x_C) \in \mathbb{R}^{G\times C}$, while the estimate of RNA velocity matrix is $\hat{\boldsymbol{V}} = (v_1,\dots,v_C) \in \mathbb{R}^{G\times C}$. The procedure that will be now described is repeated R times, in order to stabilise as much as possible the inferred network. A new expression matrix $\tilde{\boldsymbol{X}}$ and a new velocity matrix $\tilde{\boldsymbol{V}}$ are generated in two steps.

1. A random sample of $\lfloor C/2 \rfloor$ cells are randomly sampled from $\boldsymbol{X}$ and $\hat{\boldsymbol{V}}$: the dimensions of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{V}}$ are reduced to $G \times \lfloor C/2 \rfloor$;
2. Each row i of $\tilde{\boldsymbol{X}}$ is multiplied by a factor β_i, which is sampled from a uniform distribution $\mathcal{U}([\alpha,1])$.

For each generated pair $(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{V}})$, the matrix $\boldsymbol{A}$ is estimated by solving a lasso regression problem, defined as in Equation 4.3.

$$\min_{\boldsymbol{A}\in\mathbb{R}^{G\times G}} \|\tilde{\boldsymbol{V}} - \boldsymbol{A}\tilde{\boldsymbol{X}}\|_2^2 + \lambda\|\boldsymbol{A}\|_1 \tag{4.3}$$

The parameter λ is chosen among a grid of regularisation values, from 0 to L. It ensures that the predicted regulatory network has at least λ non-zero entries in each row. It represents the fact that the j-th gene is among the top λ transcription factors regulating the i-th gene.

This entire procedure is iterated R times. During the steps, for each pair (i,j) of genes and for each $l \in [1,L]$, a frequency $F(i,j,l)$ is computed as the amount of times that the entry a_{ij} of $\boldsymbol{A}$ is non-null. Then, always referring to [18], an area score is computed (Equation 4.4).

$$s_{\text{area}}(i,j) = \frac{1}{L}\sum_{l=1}^{L} F(i,j,l) \tag{4.4}$$

Even if the area score is the default choice in GRISLI and it has be proven to be more stable [18], a variation of this score is provided. It is referred to as the original stability selection score

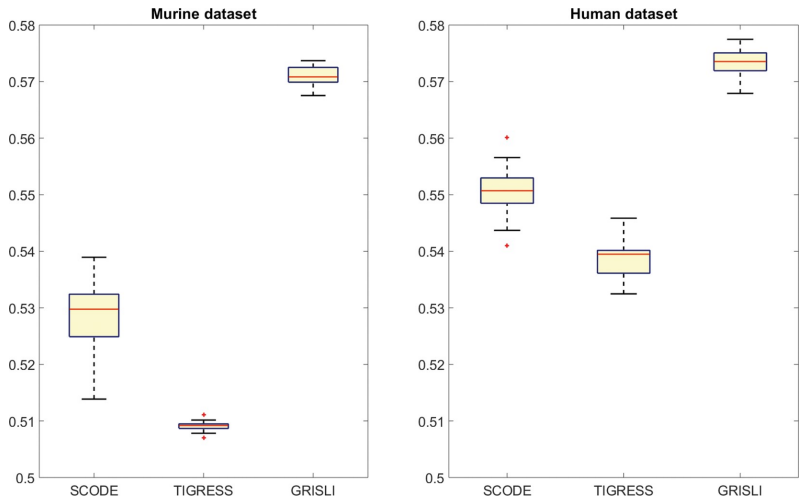


Figure 4.3: Image from [6]. Boxplots show the performances of the three methods over 30 iterations. Two datasets are examined: the murine dataset on the left [7] and the human one on the right [8]. GRISLI stably outperforms both SCODE and TIGRESS methods.

and it is defined in Equation 4.5, proposed by [19].

$$s_{\text{original}}(i,j) = F(i,j,L) \tag{4.5}$$

The hyper-parameter selection is a tough and delicate problem. In [6], they suggest that R should be the largest possible, meaning that the number of iterations should be big enough to reduce random fluctuation of the predicted GRN. The value of α should be chosen in the range $[0.2, 0.8]$ and L should be verified on a large interval of values. These two last parameters are found to strongly depend on the considered dataset. The effect of α can increase the diversity between the single batches. On the other hand, L limits the number of edges that can be assigned to the estimated regulatory matrix and its value is related to how much the GRN is expected to be sparse.

With respect to SCODE method, from which it takes the dynamical model, GRISLI has many innovations. First of all, the velocity inference step avoid the numerical integration of the left-hand side of Equation 4.2. Then, the assumption that all cells lie on the same trajectory is not fundamental anymore. Finally, no restriction on the mathematical properties of $\boldsymbol{A}$ is made and GRISLI is still able to solve a convex problem to estimate the GRN, in an efficient way from a computational point of view.

Even if the authors found GRISLI to outperform SCODE in terms of AUROC, as noticed

in [13], its performance is again poor. They tested GRISLI on two datasets and compared the AUROC scores to the ones obtained by SCODE and TIGRESS. The first data comes from the reprogramming murine embryonic fibroblast cells to myocytes [7]; the second one from the differentiation of human embryonic stem cells to definitive endoderm cells [8]. Figure 4.3 shows a boxplot for both the datasets and it clearly indicates a net improvement for GRISLI in terms of AUROC. However, recalling that 0.5 is the threshold of a random generated network, the performance is just above 0.57 and the predicted matrix cannot be considered as really meaningful, as remarked in [13].

5 Imputation and simulation methods

The topic of this Chapter consists in two fundamental tools that have been used during the thesis internship. In Section 5.1, a general discussion about imputation methods is proposed and two methods in particular are examined. The Section 5.2 addresses the simulation tools for single cell data and their challenges, and finally, two simulators are reviewed.

5.1 Imputation methods

As has been said before, single cell RNA sequencing data is characterised by a high percentage of zero values, caused by two main reasons. The first one is the limit of the sequencing technologies and the second is the fact that not all the transcriptional processes may be activated during the experiment. In general, two types of zero values can be distinguished. The *true zeros* are generated by genes that are not truly expressed. Then, the *dropout zeros* are missing values caused by too low mRNA abundance or by the stochastic pattern of gene expression at the single cell level [10].

The sparsity of the data matrix could be a further challenge for most of the analyses performed on single cell data. Moreover, there is not a standard preprocessing workflow to address these dropout events, as it exists for bulk data. Indeed, it is easier to impute the zero values when considering bulk data, as it is the result of the average expression among a large cell population. Many methods have been developed for the imputation of missing values in bulk RNA-seq data and they can be divided into five different strategies [10]. The first imputation class consists

in averaging gene expression at a gene or cell level. Then, the k-Nearest Neighbours technique can be adapted to achieve an estimation of null values from similar entries, through similarity metrics among genes. Some statistical modelling can be exploited and can perform imputation of missing values, and other methods execute many iterations and give as output a combination of the results. Finally, a more biological approach can be used, as some methods exploit information such as gene ontology to facilitate the imputation.

However, a discussion similar to Chapter 4 can be addressed: these methods implemented for bulk data may not have the same performances when applied on single cell data. In the first place, scRNA-seq data presents a larger cell-level variability than bulk data and the latter has a much smaller percentage of missing values. Nevertheless, the main and most important complication is that dropout events in sc-data are given by a mixture of dropout zeros and real zeros.

These reasons brought the scientific community to develop new imputation methods only for single cell data. In subsection 5.1.1 and 5.1.2 two methods implemented specifically for scRNA-seq data will be discussed in detail.

A few preprocessing techniques can be anticipated, which are standard for most of the imputation methods. First of all, the data matrix should be the count matrix, with only integer entries, that can be obtained from standard sequencing methods. Then, the data is normalised with respect to the library size of each cell, that is defined as the total number of sequenced RNA reads. After that, the logarithm function is applied, after having added a pseudocount (1.01 for scImpute method and 1 for DrImpute), in order not to have infinite values due to the logarithm of zero entries. These two steps are useful on one hand because the data values become continuous, making most of the methods more efficient. On the other hand, the logarithmic transformation prevents the large measurements of RNA abundance to be excessively influential.

Another method that is common to be applied is the Principal Component Analysis (PCA). It is performed on the transformed data matrix and reduces its dimensions along the genes, preserving the cells number dimension, and identifies the Principal Components (PCs). This procedure helps to diminish the negative effect of frequent dropout events.

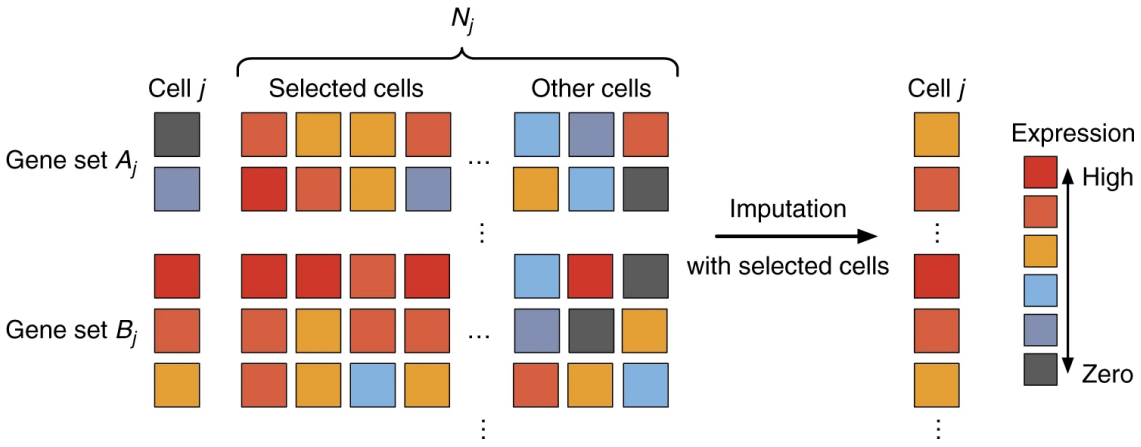


Figure 5.1: Figure from [9]. The scheme shows the workflow of scImpute method. Low expressed genes with high probability of being dropout events in cell j are imputed (gene set A_j). A subset of the other cells N_j is selected based on the gene set B_j, which is not affected by dropout. The imputation is performed on the basis of the gene expression of those selected cells. The result is represented in the right part of the scheme, with the vector cell j that is changed only on the upper entries corresponding to the imputed gene set A_j, while the lower entries that are not affected by the imputation process.

5.1.1 scImpute method

The first method that can be discussed is *scImpute* [9]. It is a statistical method that imputes zero values, first by identifying the more likely dropout and then it estimates only those values, without adding any noise to the rest of the data. The reason is that, as mentioned before, not all null-values are caused by dropout events, and thus, not all of them should be imputed.

The first step of the method is to learn the probability of dropout for each gene in each cell, based on a statistical mixture model. The second step is the actual imputation of those zeros with highest likelihood to be derived from dropout events. The expression of the same gene of other cells is acquired and used during the imputation process. This information comes only from cells that are less likely to be affected by dropout. A schematic workflow of scImpute is represented in Figure 5.1.

In detail, the input of the method is a count matrix X of dimensions $G \times C$, meaning the rows represent the genes and the columns the cells. A normalisation preprocessing is applied with respect to the library size of each cell and then it is transformed with a logarithm function, as shown in Equation 5.1.

$$X_{ij} = \log_{10}\left(X_{ij}^{N} + 1.01\right) ; i = 1, \ldots, G,\ j = 1, \ldots, C \tag{5.1}$$

The second step is identifying clusters of similar cells from which to borrow genes information. Since the percentage of null values is high, it is difficult to detect the true cell types and thus, only candidate neighbours can be selected for each cell. The PCA is performed on the data matrix X and the newly generated matrix Z has the PCs as rows. Then, a matrix $D_{C\times C}$ is computed as the distance between the cells. Now outlier cells must be identified. From this matrix, a list L is created and each value is the minimum distance of a cell from its neighbours: $L = \{l_1, \ldots, l_C : l_j = \min D_{C\times C}(\cdot, j)\}$. The outliers are identified following the common definition (Equation 5.2), where Q_1 and Q_3 are the first and second quantile for L.

$$O = \{j : l_j > Q_3 + 1.5(Q_3 - Q_1)\} \tag{5.2}$$

The set of candidate neighbours for the outlier cells is set to the empty set, $N_j = \emptyset$. It should be noted that these outlier cells may be the result of technical errors or may correspond to a rare cell type that is truly present in the experiment. However, these cells do not undergo the imputation process and are not taken into consideration as neighbours of other cells. The last step consists in the clustering of the remaining cells, $\{1, \ldots, C\} \setminus O$, into K groups. K is a hyperparameter that must be given as input by the user. Finally, a value $g_j = k$ is assigned to cell j if it belongs to cluster k. So, the candidate neighbours set is defined as $N_j = \{j' : g_{j'} = g_j,\ j' \neq j\}$.

Afterwards, a statistical model can determine if a zero entry of X is caused by dropout event or not. The model is given by a mixture of two components, as most genes express following a bimodal distribution across similar cells. The first distribution that is used is a Gamma distribution, that accounts for the dropouts and the second one is a Normal distribution, to account for the gene expression level. The parameters of the mixture of the two distributions are specific for each cluster k. For each gene i and each subpopulation k, the expression is defined as a random variable $X_i^{(k)}$ that follows the density distribution defined in 5.3. The parameter $\lambda_i^{(k)}$ is the dropout rate of gene i within the cluster k and the other parameters describe the distribution they are associated with.

$$f_{X_i^{(k)}}(x) = \lambda_i^{(k)} \text{Gamma}\left(x; \alpha_i^{(k)}, \beta_i^{(k)}\right) + \left(1 - \lambda_i^{(k)}\right) \text{Normal}\left(x; \mu_i^{(k)}, \sigma_i^{(k)}\right) \tag{5.3}$$

This model represents the idea that, if a gene has high expression level and low variation among the subpopulation, then a null value is more likely to be an effect of a dropout event. Vice versa, if the medium expression is low and the variation is high, then the zero may be assumed

to reflect a real biological behaviour.

At the end of these steps, the estimation of the parameters in Equation 5.3 is obtained by the Expectation-Minimisation algorithm and the dropout probability of gene i and cell j is given by Equation 5.4.

$$d_{ij} = \frac{\hat{\lambda}_i^{(k)} \text{Gamma}\left(X_{ij}; \hat{\alpha}_i^{(k)}, \hat{\beta}_i^{(k)}\right)}{\hat{\lambda}_i^{(k)} \text{Gamma}\left(X_{ij}; \hat{\alpha}_i^{(k)}, \hat{\beta}_i^{(k)}\right) + \left(1 - \hat{\lambda}_i^{(k)}\right) \text{Normal}\left(X_{ij}; \hat{\mu}_i^{(k)}, \hat{\sigma}_i^{(k)}\right)} \tag{5.4}$$

Finally, the imputation step is performed cell by cell. A gene set A_j identifies the genes with high probability of being dropout in cell j and it is defined as $A_j = \{i : d_{ij} \geq t\}$. The complementary gene set $B_j = \{i : d_{ij} < t\}$ is used to select those cells that are similar to cell j, by using a non-negative least square regression (Equation 5.5).

$$\hat{\beta}^{(j)} = \text{argmin}_{\beta^{(j)}} \|X_{B_j,j} - X_{B_j,N_j}\beta^{(j)}\|_2^2, \quad s.t.\, \beta^{(j)} \geq 0 \tag{5.5}$$

This type of regression has the property of giving a sparse estimate, with exact zero components, so the parameter $\hat{\beta}^{(j)}$ is used to select cells which are similar to cell j, among its neighbours set N_j. The final step is the actual imputation of gene expressions in the set A_j (Equation 5.6).

$$\hat{X}_{ij} = \begin{cases} X_{ij}, & \text{if } i \in B_j, \\ X_{i,N_j}\,\hat{\beta}^{(j)}, & \text{if } i \in A_j. \end{cases} \tag{5.6}$$

5.1.2 DrImpute method

The second method that can be introduced is *DrImpute* [10]. It is a deck imputation approach developed specifically for scRNA-seq data. A hot deck imputation method handles the putative missing data by replacing them with measurements belonging to a somehow similar unit.

The first step of this method is the identification of similar cells on the basis of their gene expression, by using clustering techniques. For this reason, DrImpute is defined as a deterministic method, since the selection of similar units is not randomly performed. Then, the average expression of similar cells is used to impute all the null values of a specific cell. This imputation is iterated many times in order to achieve some sort of stability, by using different clustering results. The final imputed data matrix is the average of the many results obtained from different iterations (Figure 5.2).

Figure 5.2: Figure from [10]. Trivial example of DrImpute worflow. The first step is cell clustering (left). Within each identified cluster, imputation process of zero entries is performed (right). The imputed values (red numbers) are obtained as the average of the gene expression levels of the similar cells.

One limitation of DrImpute is that the imputation is based only on cell-level correlation and on a relatively simple hot deck approach. Indeed, the gene-level correlation should also be considered and modelled, since it exists and it is already considered in imputation methods for bulk RNA-seq data. This improvement could lead to better performance if implemented.

It has been shown in [10] that the performance of many statistical tools improves when the imputed data is provided as input. Moreover, it has been proven to impute in a better way than scImpute does.

Going into details, DrImpute needs as input only the preprocessed data matrix $\tilde{X}$ and the number of clusters that is assumed to characterise the cell population. The matrix $\tilde{X}$ must have dimension $G \times C$, where G is the number of genes and C the number of cells. The preprocessing is standard and also similar to that performed by scImpute. The counts matrix is normalised by size factor and so the matrix X^N is obtained. A logarithmic function is then applied, after adding a pseudocount of 1 to the normalised matrix, in order to avoid errors when computing the logarithm of a null value (Equation 5.7).

$$X_{ij} = \log_{10}(X_{ij}^N + 1)\,; i = 1, \ldots, G,\ j = 1, \ldots, C \tag{5.7}$$

Within the algorithm, the parameter H represents the number of clustering configurations that are explored. It is the result of various combinations of distance metric functions applied on the range of number of clusters that is given as input. Each combination provides a clustering configuration, denoted as $C_1, \ldots, C_H$. In particular, the default clustering approach is similar to that of SC3 method [20]. Two similarity matrices among cells are generated, using Pearson and Spearman correlations. Then, K-mean clustering is performed only on 5% of

the principal components retrieved from the similarity matrices. The default range of clusters goes from 10 to 15 groups. In this case, the total number of configurations is $H = 12$, since two metrics and 6 numbers of clusters are considered. The user can of course change both the distance construction methods and the number of subgroups, according to their assumptions about the studied cell population.

Iteratively, each clustering result C_h is assumed to be a true cell classification and the imputed value of a dropout event can be inferred as the average values among the cell cluster (Equation 5.8).

$$\mathbb{E}(X_{ij}|C_h) = \text{mean } (X_{ij}|X_{ij} \text{ in the same group in clustering configuration } C_h) \tag{5.8}$$

The final imputation is obtained by averaging only the presumed dropout events X_{ij} among the different clustering results $C_1, \ldots, C_H$, as described in Equation 5.9.

$$\mathbb{E}(X_{ij}) = \text{mean } (\mathbb{E}((X_{ij}|C)) = \frac{1}{H}\sum_{h=1}^{H}\mathbb{E}(X_{ij}|C_h) \tag{5.9}$$

5.2 Simulation methods

It is a common approach to evaluate the performance of computational tools, which are run on synthetic generated datasets, considered as the ground-truth data. The simulators should resemble real data features and their usage is crucial in the assessment phase, to measure quality and robustness of the methods. The main objective of the simulation tools is to maintain the biological signalling properties, while keeping in mind the computational applicability too. The exponential growth of analysis tools for scRNA-seq data have made it necessary to develop high-performing simulators. They should provide in silico data with a user designed structure and groutruth parameters, such as the number of cell groups.

In the literature, most simulation tools involve two steps. The first one is the definition of a statistical model which describes the characteristics of real scRNA data. Then, this information is used as a template to actually simulate data, that is then given as output. The first developed methods use the Negative Binomial (NB) distribution for the gene expression modelling. This distribution is proved to provide simulated data whose variance accurately resembles that of real data. The Poisson distribution has been also taken into account, but it

requires a further assumption (that the mean and variance are equal) that is usually not met in real experiments. One of its variants, the zero-inflated NB, has been considered to obtain data with a better sparsity configuration. Recently, other models have adopted a mixture of two statistical distributions to increase the flexibility of the modelling phase (as done also by scImpute in Equation 5.3). Lately, deep learning based approach exploits neural networks ability to capture the underlying gene expression distribution, avoiding any prior assumptions.

To recap, simulation tools aim to reproduce realistic datasets, characterised by both cell- and gene-wise features, as well as high-order interactions [21].

In the following subsections, two simulators are introduced. Their singularities is that, besides the simulation, they start from a given gene interaction network. Moreover, the RNA velocity matrix can be retrieved. Both of these two pieces of information are necessary for the purpose of this thesis, as they become the ground-truth for the evaluation of both velocity inference and GRN inference.

5.2.1 SERGIO simulator

Most existing simulation tools for single cell data do not take into account the gene regulatory networks that control the dynamics of expression. Indeed, transcription factors play a major role in gene dynamics and their concentration can considerably alter the target's expression. SERGIO is a simulator that models the stochastic nature of transcription, starting from information on the transcription factors which regulate the genes [22]. It can model any number of cells, provided by the user, in both steady state or differentiation trajectory.

SERGIO is focused on three main aspects. A mathematical model describes the regulatory dynamics underlying gene expression. The cell-to-cell variability, which characterises real sc-data, is modelled as a stochastic component of the method. Finally, the technical errors due to the sc-technologies are incorporated and can be applied on the expression matrix generated before.

In detail, Equation 5.10 represents the stochastic differential equations that SERGIO is based on, called Chemical Langevin Equations (CLE). They simulate the gene dynamics of unspliced and spliced RNA as a function of the TFs levels, by following the GRN interactions between them. The variables u_i and s_i are the expression of gene i at unspliced and spliced level respectively, λ_i, μ_i and γ_i represent the degradation of unspliced, splicing and degradation of spliced rates and the term q_i acts as noise amplitude in the transcription levels. The parameters

α, β, ϕ and ω are independent Gaussian processes, needed for the addition of white noise. The parameter P_i is the production rate of gene i. It is calculated as the sum of the effect of all regulators on the gene i.

$$\begin{aligned} \frac{du_i}{dt} &= P_i(t) - (\lambda_i + \mu_i)\, u_i(t) + q_i^u \left(\sqrt{P_i(t)}\alpha + \sqrt{(\lambda_i + \mu_i)u_i(t)}\beta \right) \\ \frac{ds_i}{dt} &= \mu_i u_i(t) - \gamma_i s_i(t) + q_i^s \left(\sqrt{\mu_i u_i(t)}\phi + \sqrt{\gamma_i s_i(t)}\omega \right) \end{aligned} \tag{5.10}$$

The CLE is advantageous mostly because numerical integration methods can be used to derive the mRNA concentrations of genes in a computationally efficient way.

SERGIO allows the addition of technical errors, which are of three types. The first one is the outlier genes phenomenon. It refers to proven observations that few genes have usually an oddly high expression level among cells. A hyper-parameter must be given as input to define the probability of a gene to be an outlier. Once selected as an outlier, its expression value is multiplied by a coefficient that follows a log-normal distribution.

$$\begin{gathered} \forall i \in \{1, \ldots, G\} : \mathbb{I}_i^O \sim Ber(\pi^O)\,, f_i^O \sim lnN(\mu^O, \sigma^O) \\ \forall i \in \{1, \ldots, G\},\, \forall c \in \{1, \ldots, C\} : x_i^c \leftarrow \mathbb{I}_i^O f_i^O x_i^c + (1 - \mathbb{I}_i^O x_i^c) \end{gathered}$$

Parameters G and C represent the number of genes and cells that are simulated, while x_i^c is the gene expression that can be either unspliced or spliced. The variable $\mathbb{I}_i^O$ is sampled from a Bernoulli distribution and indicates if the gene is an outlier or not. The coefficient f_i^O is parametrised as a log-normal random variable, whose parameters are user-defined.

The second type of technical error which can be simulated is the library size parameter. For each cell, a value L_c is sampled from a log-normal distribution, characterised by mean μ^L and variance σ^L defined by the user. The gene expression is then multiplied by L_c and normalised by the total depth of all the genes in the cell.

$$\begin{gathered} \forall c \in \{1, \ldots, C\} : L_c \sim lnN(\mu^L, \sigma^L) \\ \forall i \in \{\ldots, G\},\, \forall c \in \{1, \ldots, C\} : x_i^c \leftarrow \frac{L_c}{\sum_{j \in \{1,\ldots,G\}} x_j^c} x_i^c \end{gathered}$$

Finally, the last and most important phenomenon that can be simulated is the dropout effect. For each cell, a probability is assigned to each gene, corresponding to the likelihood of being subject to dropout. The authors make a strong assumption here: the more the gene is expressed,

the less likely it is to be cancelled. For this reason, the probability of dropout is modelled as a logistic function. This value is then used as the parameter of a Bernoulli distribution, which is used to decide whether the gene is affected by dropout or not. In particular, unspliced and spliced abundances are affected independently by dropout and values are sampled in parallel.

$$y_0 = q^{th}\text{percentile of } Y, \text{where } Y = \log(X+1), X = \{x_i^c\}$$
$$\forall i \in \{1,\dots,G\}, \forall c \in \{1,\dots,C\} : \pi_{i,c}^D \leftarrow \frac{1}{1+\exp\left(-k(Y_{i,c}-y_0)\right)}\,,\ \mathbb{I}_{i,c}^D \sim Ber(\pi_{i,c}^D)$$
$$x_i^c \leftarrow \mathbb{I}_{i,c}^D x_i^c$$

5.2.2 DYNGEN SIMULATOR

The second simulation tool is *dyngen*, a method that is based on a multimodal representation [23]. It is a three-step method that can be configured directly by the user. The biological transcriptional processes are obtained starting from regulatory interactions, that are translated into the set of reactions of regulation, transcription, splicing and translation. Then, each cell is simulated separately from the others, by exploiting a stochastic simulation algorithm, called Gillespie's SSA. Finally, some real datasets are provided and, once the user selects one of those, it is used to emulate the corresponding sc-profiling protocol.

The interesting property of dyngen is that it provides a large variety of differentiation trajectories, such as bifurcating or cyclic, and many experimental conditions, for example time-series and perturbations. In addition, the user can set two parameters so that the method gives as output both the gene regulatory network and the matrix of RNA velocity.

In detail, the first thing that is defined is the module network. A module network is modelled by regulatory interactions that can be up- or down-regulating, and defines the trajectory of differentiation of the simulated cells. dyngen provides different modules, from the simplest linear process, to the bifurcating or cyclic. Also, more than one of these chains can be selected and concatenated, creating a module with longer chains.

According to the selected module, a GRN is generated in four main steps. Transcription factors are individuated and their interactions are set. A number of target genes, provided as input, is added to the regulatory network and finally, the so-called "housekeeping" genes are simulated. The latter are genes that are always highly expressed, independently from both transcription factors and the other genes. They usually codify for proteins that are fundamental

for cell survival.

The next step is the translation of the GRN to the set of reactions, based on a stochastic framework. For each gene, the abundances of pre-mRNA, mature mRNA and protein are tracked. Then, the reactions of transcription, splicing, translation and degradation are defined and the expression changes mimic the real biological effect. For example, if the splicing reaction considers one molecule of pre-mRNA, then a new molecule of mature RNA is generated. Except for the transcription reactions, the propensity of the other reactions are regulated by linear dependencies. The transcription process is defined by a more complicated model, derived from thermodynamic models of gene regulation. The main concept is that a promoter of a gene can be bound or not by some transcription factors and the propensity can be computed through nonlinear functions.

One of the most crucial stages is the generation of single cell dynamics. The SSA simulation consists of multiple iterations, where at each step one reaction is triggered. An external library is used for performing those simulations in an efficient manner.

Once the SSA simulations are performed, all the gene expression levels are available for each state. The technical effects are now introduced in order to get close to the real data patterns. The library size and dropout effects are derived from the real dataset that the user selected during the initialisation of the module.

6

Evaluation of the GRISLI algorithm

The purpose of this thesis is the investigation of the GRISLI algorithm developed in [6]. It is based on the linear ordinary differential equation (6.1) and aims to reconstruct the gene regulatory network that characterises a given scRNA-seq dataset.

$$\frac{dx}{dt} = Ax \tag{6.1}$$

The left-hand side of the equation represents the RNA velocity, expressed as a ratio between the change in gene expression and the corresponding infinitesimal change in time. On the right-hand side, the matrix A is associated with the GRN and characterises how the expression of one gene influences the dynamics of all the others.

Briefly recalling its workflow, this method is divided into two main steps. Firstly, instead of integrating the left-hand side of Equation 6.1,it infers the RNA velocity from the spliced counts matrix through the three-point model described in Section 3.4, exploiting the pseudotime information. Then, using the expression and velocity matrices, the lasso regression problem is formulated and solved many times, in order to achieve a GRN prediction that is as stable as possible. At each iteration, a frequency score is associated with each entry of the GRN matrix and the final output of the method is the combination of all the results obtained during these repetitions, according to that score.

In Section 6.1 some initial analyses are carried out, to study in depth the choices of GRISLI authors and the relationship between genes and their regulators. Section 6.2 is the main core of

this work and presents the results and limitations about velocity inference methods, with the main focus on scVelo. Finally, Section 6.3 studies the effect of imputation as a preprocessing step applied on both simulated and real data.

6.1 Preliminary analysis

Before starting any analysis, it must be said that GRISLI is implemented as a MATLAB algorithm. In order to make it more readable, the beginning of the internship has been dedicated to its translation in the more accessible python language. This period of time has allowed us a deep understanding of the whole method and the main computational passages.

The first objective of the internship is to investigate and reproduce the analysis performed in [6]. Since one main step of the algorithm involves the resolution of a linear regression problem, other computational methods are explored. In particular, the original method is the `lasso` function provided by the `SPAMS` (SPArse Modelling Software, v2.6) package, developed for both Matlab and python languages. The formulation of the minimization problem is recalled in Equation 6.2, where V is the velocity matrix and X the spliced gene expression matrix.

$$\min_{A \in \mathbb{R}^p} \|V - AX\|_2^2 \quad \text{such that } \|A\|_1 \leq \lambda \tag{6.2}$$

The matrix A, that is given as output, is a sparse matrix and represents the inferred GRN.

We modify this step and investigate the performance of GRISLI using different methods. The first one is the `Lasso` function provided by the `Linear_model` module of the `scikit-learn` library. Then, the least-squares method of the `numpy` package is used. Since, as discussed in Chapter 4, the performance of GRN reconstruction methods are really poor, the performance of GRISLI is also tested on a regulatory matrix that is obtained in a random way. Finally, an investigation about the importance of the pseudotime information is analysed. At the beginning of the GRISLI workflow, the vector of cells' times is shuffled and all the downstream analyses may be affected by this manipulation.

In Figure 6.1, two boxplots are presented. The upper one shows the comparison of these different approaches in terms of AUROC. As expected, the mean AUROC of random constructed matrix performance is exactly 0.5. The box corresponding to the `SPAMS Lasso` has the best AUROC value with respect to all the others. The lower boxplot shows the computational cost in terms of time (seconds) of these approaches. It is noticeable the difference between the `SPAMS Lasso` when applied on the correct dataset and on the one with the shuffled

temporal vector. This means that, even if the performance with shuffled times is a little above the random baseline, it requires an extreme effort from the lasso regression method to find the solution of the minimisation problem.

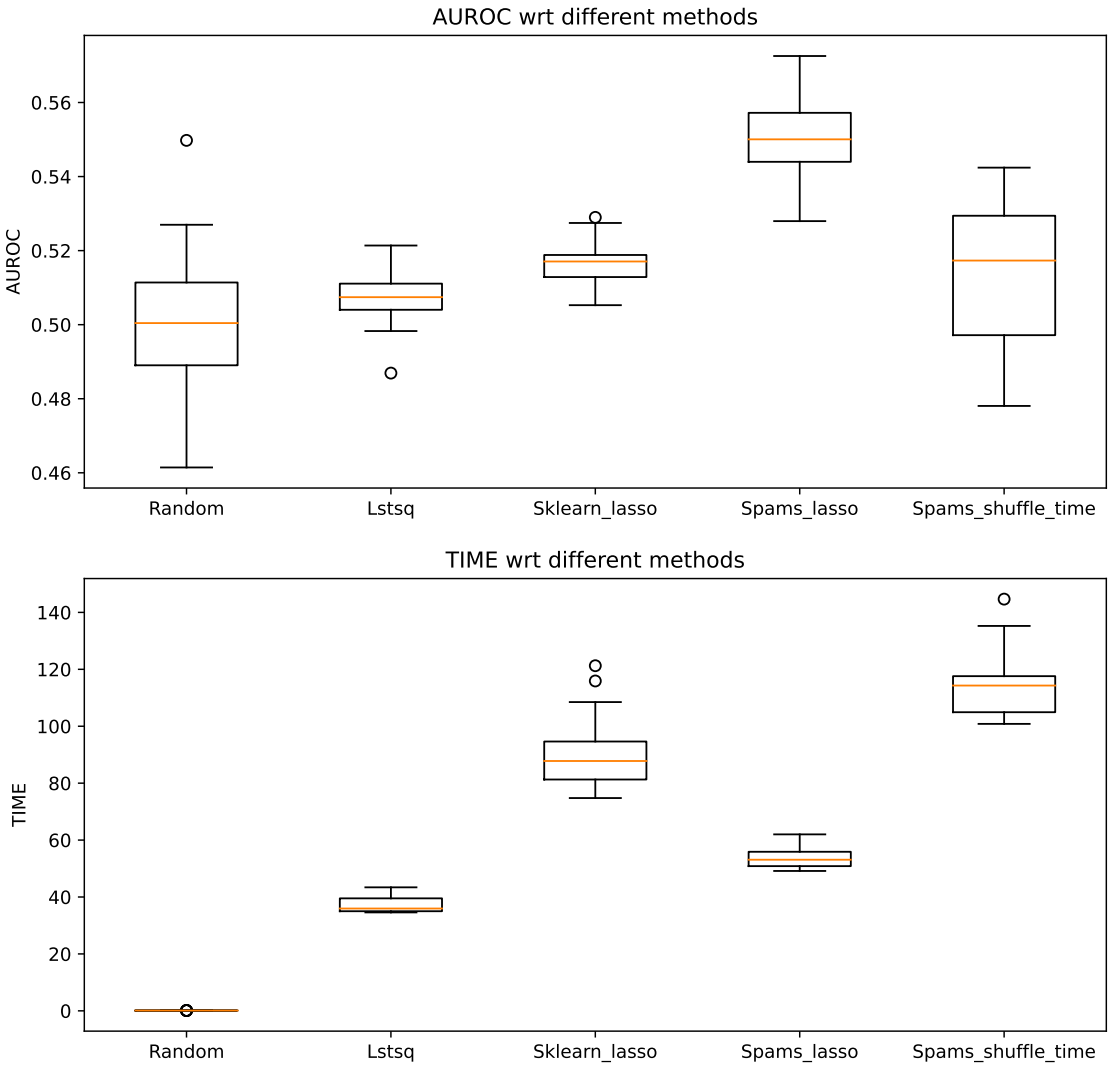


Figure 6.1: Two boxplots presenting the performance of the different methods for the minimisation problem (Equation 6.2). The upper chart shows the AUROC values for random generated GRN matrix, using least squares approach, Lasso regression of `scikit-learn` and SPAMS packages, and finally the results using shuffled pseudotime vector. Clearly, authors' choice of the SPAMS tools improves the performance. The lower chart shows the computational cost of the different approaches in terms of time requirement (seconds). These tests were performed on dataset [7], but similar boxplots are obtained for both dataset [8] and dataset [11].

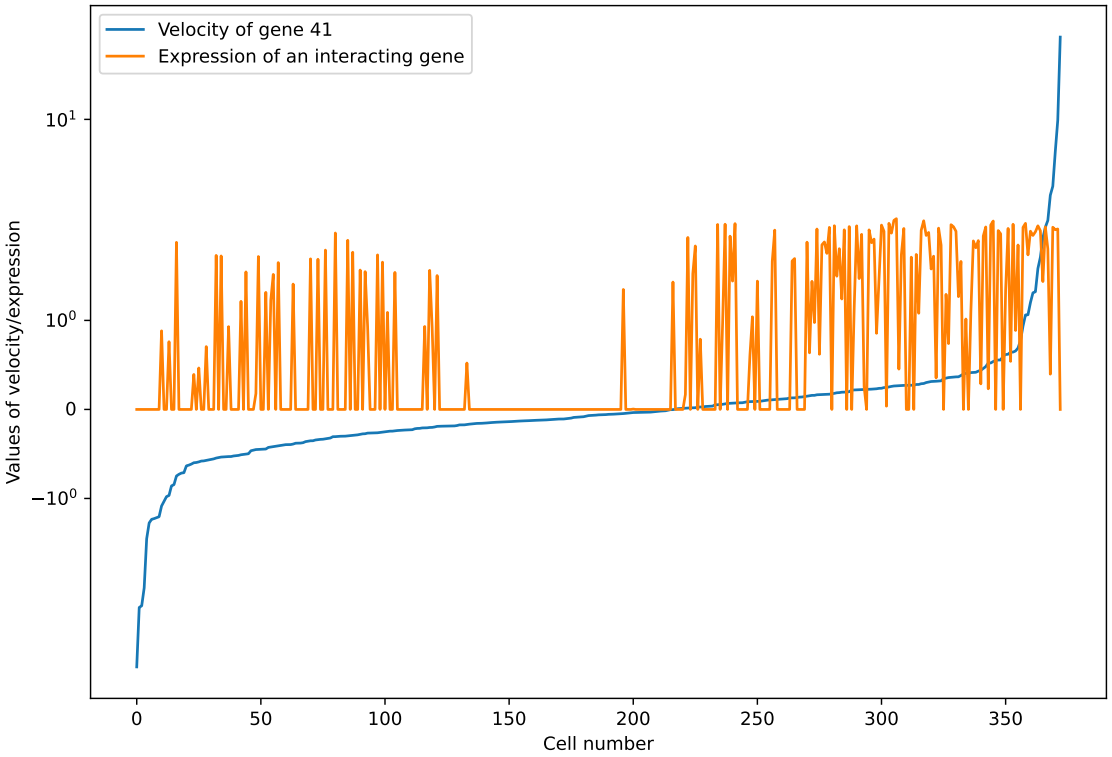


Figure 6.2: The plot shows the relation between the velocity of gene number 41 (blue line) and the expression of one of its regulators (orange line). When the latter is little expressed (left part of the graph), the target gene is in a repression phase, probably due to the effect of the other regulators. When the transcription factor is highly expressed (right part), the target gene increases its velocity. Probably, this regulator gene is an activator of the expression of the gene 41. The data used for this analysis comes from [7].

Another preliminary analysis is the study of the consistency between the expression of target genes and their regulators. In particular, if the presence of a regulator is really fundamental and has a real effect on the inhibition or activation of the final target gene. In order to examine these relationships, one of the dataset used in [6] is considered, which is obtained by the reprogramming of murine embryonic fibroblast cells to myocytes, presented in [7]. This dataset contains 373 cells and 100 genes are filtered.

The gene number 41 is selected, since it has five transcription factors as regulators. In Figure 6.2 the velocity of gene 41 is indicated by the blue line. It is given by the interpolation of the single velocity values for each cell, which have been sorted in advance to realise a continuous graph. One regulator of gene 41 is selected and its expression among all cells is shown as an orange line.

The first thing to observe is that when the gene 41 velocity is null, its regulator is silent or

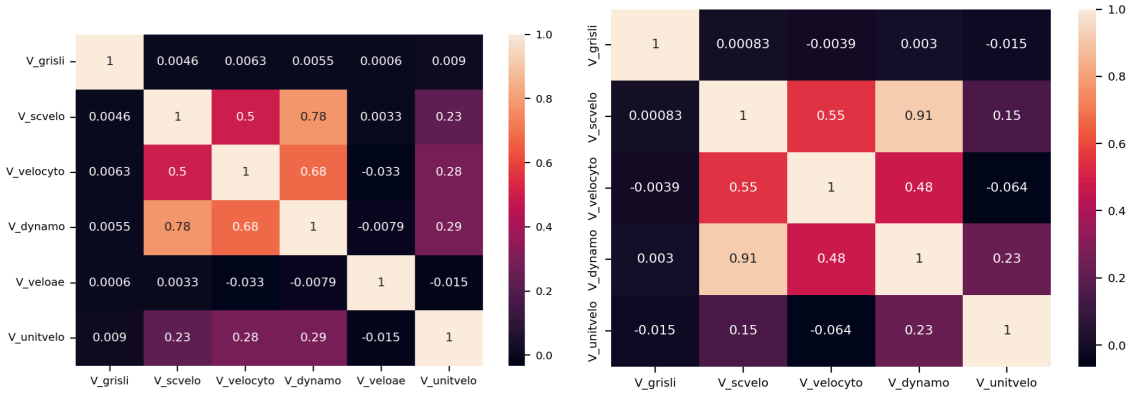
little expressed. Vice versa, when the transcription factor is highly expressed, on the right side of the graph, the velocity of the target gene increases. Even if it is not the only regulator of gene 41, the relationship between those two genes is evident.

6.2 RNA velocity inference analysis

Once it has been shown that the authors' choice about the regression method is the best among the most common ones, the next objective of the internship is the benchmarking of methods for RNA velocity. First, it must be said that in order to properly evaluate the inferred velocity matrices, a ground-truth matrix should be available to compare them with. For this reason, the simulation tools introduced in Section 5.2 are used to generate different datasets to carry out all the needed analyses. In particular, from both algorithms, it is possible to retrieve the RNA velocity underlying the expression dynamics of the single cell simulations. Now, the predicted velocity matrix can be compared to the "real" matrix and errors can be calculated and used for the evaluation of the performances.

These analyses aim to improve the prediction of RNA velocity, in order to provide more precise data to GRISLI and, hopefully, improve its performance. The three methods introduced in Chapter 3 and others more recently developed are considered:

- three-point method, implemented in the first step of GRISLI algorithm;
- scVelo, based on the two differential equations describing both unspliced and spliced mRNA dynamics;
- velocyto, also known as steady-state method, based on the strong assumption of observing both passive and active steady states of gene expression;
- dynamo, proposed by [24], based on a similar model to scVelo's one, and that, besides velocity, reconstructs also a continuous vector field of trajectories and extract gene regulations;
- veloAE, proposed by [25], an autoencoder based method that, through the projection of expression data onto a low-dimensional space, aims to denoise the velocity information;
- unitvelo, proposed by [26], a statistical framework that models the spliced RNA as a function of time and gene-specific parameters in a top-down strategy, allowing a more flexible gene expression profile.



(a) SERGIO simulations **(b)** dyngen simulations

Figure 6.3: Two heatmaps showing the Spearman correlation scores between the different velocity inference methods that are investigated. The three-point method adopted by GRISLI and veloAE are completely unrelated to any other method. Unitvelo is not very correlated to scVelo, velocyto and dynamo, which instead are highly correlated to each other. The scores are taken as the median of heatmaps given after many iterations and many simulations have been performed (at least ten for each dataset).

The correlation between all these different methods is investigated. For each dataset simulated by SERGIO and dyngen methods, the *Spearman* correlation coefficient is calculated among all the RNA velocity matrices that are predicted. The Spearman correlation is chosen because, unlike the Pearson correlation that is commonly used, it does not assume that the given data is normally distributed. The score belongs to the range $[-1, 1]$, where -1 and 1 indicate monotonic relationship and 0 implies that there is no correlation. Moreover, if the coefficient is greater than 0, it means that there is a positive correlation and so, velocity matrices are similar to each other, and vice versa, if it is smaller than 0, there is a negative relationship and the predicted velocity are dissimilar and in some sense, opposite.

In Figure 6.3, two heatmaps are shown, summarising the correlations between the different velocity methods. They are the results of computing the median values between various simulated datasets and many iterations. The left heatmap is referring to the datasets obtained from SERGIO simulator and the right one from dyngen. The rows and columns are associated with the different velocity methods, starting with the three-point method used in GRISLI. It must be said that veloAE is not applied on dyngen simulations, because it would have taken too long to run, since many iterations are required to achieve the convergence of the algorithm.

Anyway, the two heatmaps are similar to each other. In particular, two methods are proved to be not correlated to any other ones: the three-point method of GRISLI and veloAE. Their

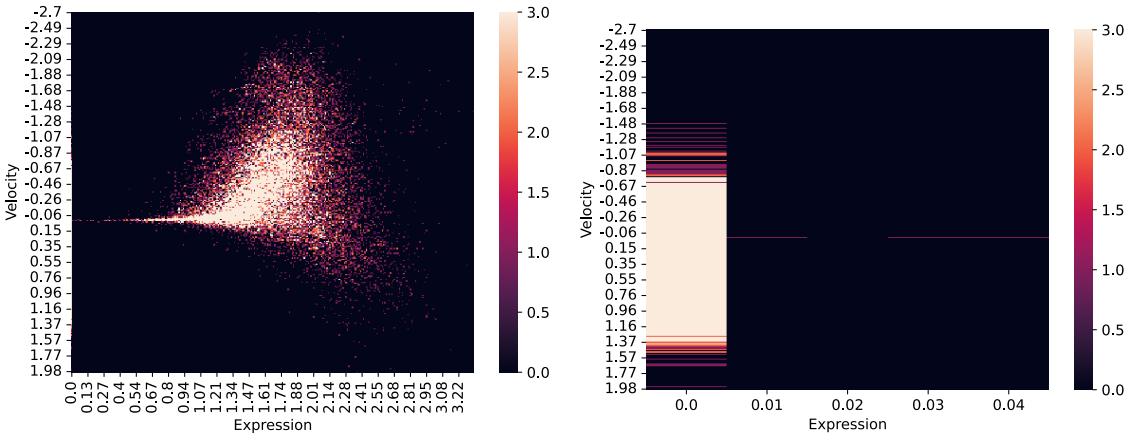
Spearman scores are always almost zero to all the other methods. In a sense, this is expected from the simplistic derivation of GRISLI velocity method, since it does not even consider the unspliced RNA expression and is based on the rate of change of spliced RNA alone. Also, an innovative method like veloAE surprisingly differs a lot from others. In all the tests performed, this machine learning model had the loss function that kept decreasing constantly, even after being running for 48 hours, completing millions of epochs. Default hyper-parameters were used and they surely could have been better adapted to each different dataset. However, the huge demand of time to be trained and the difficult parameters tuning process make the use of veloAE prohibitive.

On the other hand, three methods have high correlation scores with each other, which are scVelo, velocyto and dynamo. They all are based on the same double differential equation to describe the transcriptional dynamics and this is probably the reason why they predict velocity matrices that are highly correlated, despite different approaches being adopted. Finally, unitvelo method is little correlated to the three methods cited before, meaning that probably the statistical framework allows more flexible predictions.

Looking at the overall behaviours, except for the two methods that are not correlated with any other, the rest are characterised by positive Spearman scores, suggesting that the two differential equations modelling is consistent across different computational approaches.

In the second part of the velocity methods analysis, the relationship between gene expression and velocity is examined. In particular, during all the analyses performed, a bias towards negative velocity values has been noticed. This aspect is especially investigated for scVelo, but this tendency is observed in all the methods.

In Figure 6.4, two heatmaps show the first results of this study. They represent the spliced mRNA expression on the x-axis and the velocity values on the y-axis. From the left heatmap, the abundance of negative velocity values is immediately noticeable compared to positive ones. It is recalled that negative velocity means that the corresponding gene is being switched off and is decreasing its transcriptional phase. The right heatmap is obtained as an enlargement of the whole graph and pays particular attention to the null expression values. Indeed, while positive velocity values make sense for non-expressed genes, as they signify that the gene is being activated, the lots of negative velocity associated with silent genes is absolutely unexpected. From a biological point of view, a gene that is not expressed cannot be deactivated even further. This could be seen as a strong bias within the velocity inference methods, not just scVelo, and it should be considered as a problem to be investigated in depth.



(a) Without zoom **(b)** With zoom

Figure 6.4: The two heatmaps have gene expression values on the x-axis and the corresponding velocity values on the y-axis. There is not an evident relationship between those two variables, as expected. In the left chart, it is evident the abundance of negative velocity with respect to the positive ones. An horizontal line is also noticeable, that represents the genes with null velocity, meaning they are in a steady state phase. The right heatmap is an enlargement in a neighbourhood of 0 gene expression. It shows that scVelo assigns negative velocity even to genes that have zero expression. (In order to make the heatmap more readable, the frequency values are cut at 3 occurrences.)

An hypothesis about the lots of negative values is formulated. Since single cell data is influenced by dropout effects, i.e. expression matrices are really sparse, and these technical errors affect unspliced and spliced mRNA counts independently. Since the SERGIO simulator allows the generation of a "clean" dataset, on which technical errors can be added, an analysis about the velocity distribution is performed for different percentages of dropout.

Figure 6.5 shows the velocity distribution of the ground-truth velocity, retrieved by SERGIO, and the one inferred by scVelo. The ground-truth distribution is almost symmetric with respect to zero and the percentage of negative values is 53%. The left histogram exhibits also the distribution of scVelo velocity for the smallest amount of dropout that SERGIO allows. It is a little left skewed, with a peak that corresponds to a small positive number. With a higher parameter of dropout, this peak shifts to the left and the percentage of negative values increases to 58%. It is important to remark that this result is observed not only for scVelo, but for all velocity inference methods.

An additional analysis about this negative velocity tendency is performed, examining the role of both depth of sequencing and dropout percentage. Starting from the same "clean" dataset, a grid of combinations is constructed using several parameters for the depth and for the dropout. In Figure 6.6, the x-axis corresponds to the increasing dropout percentage and the y-axis to the

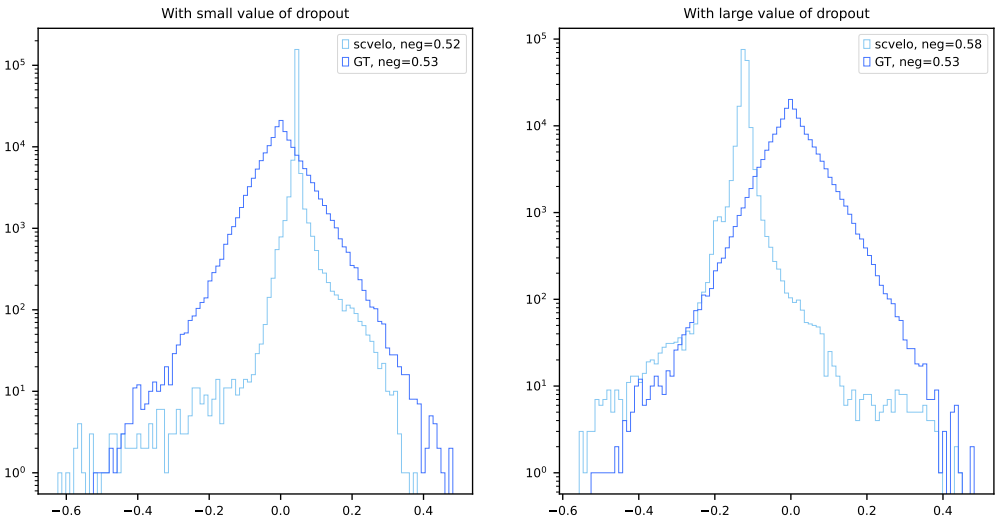


Figure 6.5: Histograms comparing the distributions of ground-truth velocity matrix (dark blue) and that inferred by scVelo (light blue). The top right box shows the fraction of negative velocities. The left histogram shows the distributions referred to a simulated dataset characterised by small percentage of dropout. The right histogram refers to a simulated dataset with a high value of dropout. The peak of the inferred velocity distribution is shifted on the left, i.e. on negative values, supporting the hypothesis that the dropout is the cause of the abundance of negative velocities.

library size. For each combination, the difference in number is calculated between the negative values of the ground-truth velocity and the inferred ones. The heatmap shows that, as the dropout percentage increases, the negative velocity values increase as well, despite the library size. This is further evidence that the high dropout is the main cause of the bias of velocity methods towards negative velocities.

6.3 Imputation of data

After observing that the velocity inference is misled by the amount of dropout that is present within the single cell data, a possible solution to limit this problem could be to impute the data as a further preprocessing step, before starting any analysis. As mentioned in Section 5.1, there are many methods for this purpose, but only scImpute and drImpute have been examined during the experiments.

The objective of this Section is to investigate whether reducing null entries through the imputation of sc-data could improve the velocity inference step and eventually the overall performance of GRISLI. The first analysis is performed on simulated data, in order to test if the hypothesis is well founded and reasonable.

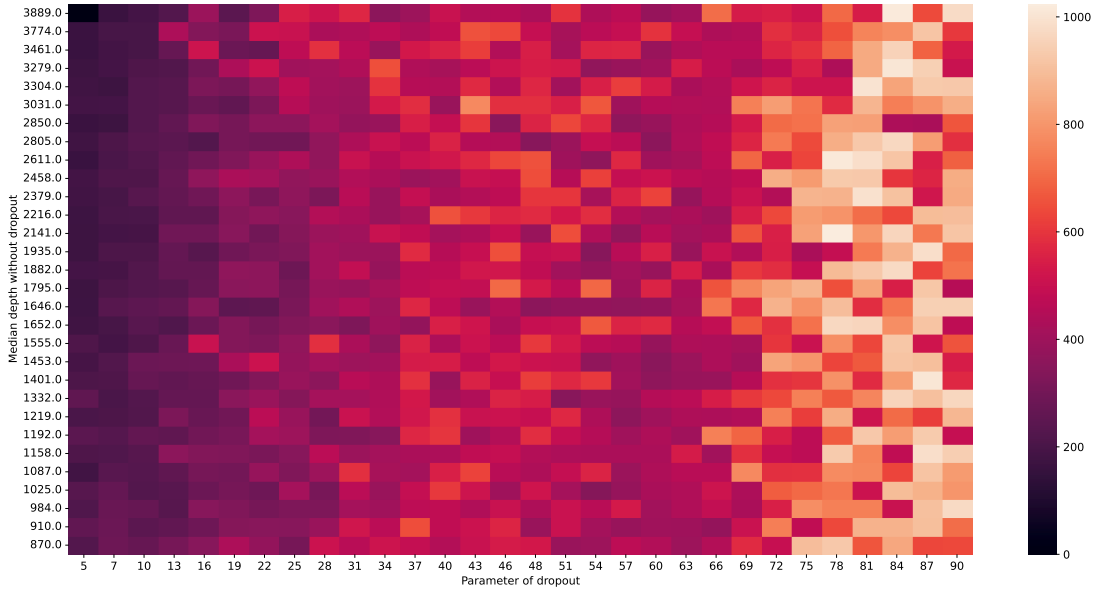


Figure 6.6: Heatmap with increasing dropout percentage on the x-axis and increasing depth values on the y-axis. The single cells correspond to the difference in number of negative velocities with respect to the ground-truth matrix. As the dropout increases, the bias towards negative velocities grows, regardless the value of depth.

The imputation can be performed on only one or both the gene expression matrices. It would be interesting to see how much this combination can affect the performance of GRISLI. For this reason, the method is applied on the original dataset, on the one where only the unspliced mRNA matrix is imputed, on the one where only the spliced matrix is imputed and on the last one where both of them are imputed. In addition, the three-point method velocity implemented within GRISLI is not used, instead the one inferred by scVelo is considered.

In Figure 6.7, the boxplot is showing the results of these investigations, that are repeated at least 30 times per sample to achieve an acceptable stability. It can be noticed that the imputation process is useful in improving the AUROC score by almost a hundredth. In particular, the imputation of the unspliced matrix affects the final results the most. It could be explained by the fact that unspliced mRNA data is usually more sparse and with shorter depth of sequencing than spliced data. The imputation may relieve this problem, increasing the information on average and thus, allowing for a better estimate of RNA velocity. Eventually, when considering both imputed matrices, the performances are also improved with respect to the original ones. Surprisingly, GRISLI performs slightly worse when the only spliced data is imputed. It can be caused by the fact that imputation does not add any additional information that could be

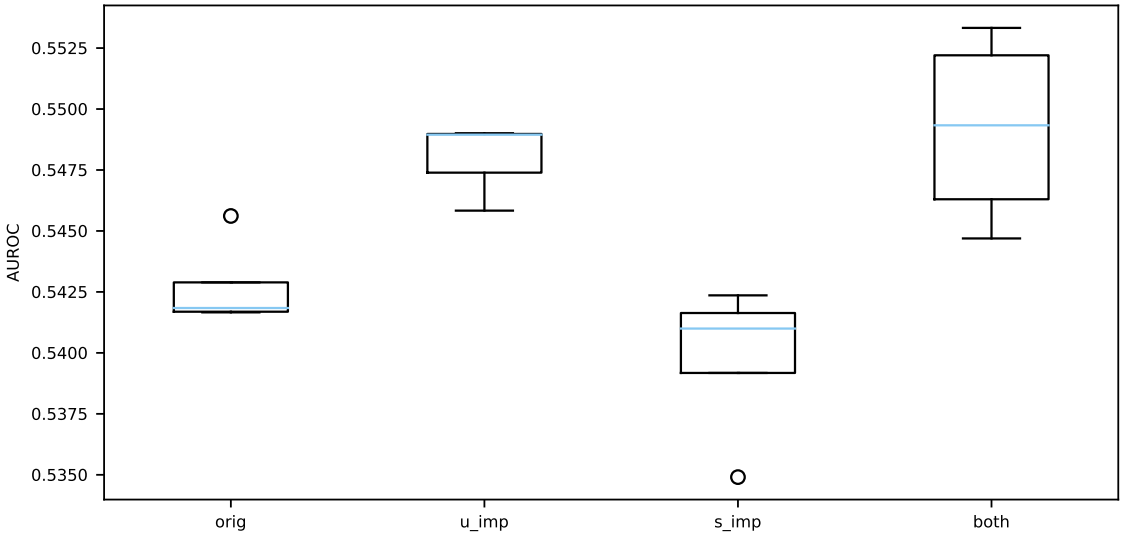


Figure 6.7: The boxplot shows the different performance in terms of AUROC for different combination of imputation on the two gene counts matrices. The tested dataset is simulated with SERGIO method and the ground-truth GRN is retrieved from the data provided as input. The velocity is computed by scVelo package. For the imputation of data, dyngen package has been used. Starting from the left, there are the box of the performance using the original matrices, then imputing only the unspliced matrix, imputing only the spliced counts and then using imputation on both unspliced and spliced matrices. There is a clear improvement in terms of AUROC score, mainly due to the imputation of the pre-mRNA data matrix.

exploited by the velocity inference step.

This overall behaviour is also found for the other velocity inference methods, consolidating the hypothesis that working on the unspliced matrix to reduce the percentage of dropout as much as possible could be a fundamental step in velocity inference.

Finally, this same analysis is tested on the real single cell data provided by [11], that is about murine pancreatic cells during pancreatic endocrinogenesis. The imputation step is executed by drImpute and scVelo is used to compute the RNA velocity matrix. Then, the GRN inference step of GRISLI is performed and the AUROC is calculated with respect to the ground-truth GRN used in the original work [6].

Figure 6.8 shows the results obtained on the real single cell dataset. The boxplot is similar to the one obtained before, displaying an improvement in terms of AUROC when the pre-mRNA matrix is imputed, and consequently when both matrices are. Interestingly, also in this case, the imputation of only the spliced count matrix seems not to be as effective as when the unspliced matrix is imputed.

Overall, when the imputation is added to the preprocessing phase and the velocity inference

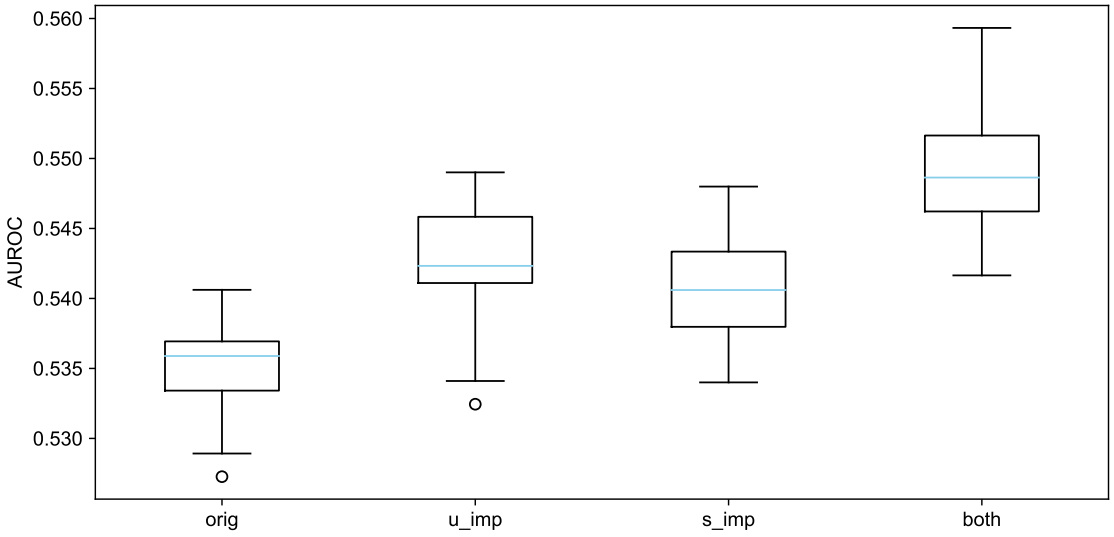


Figure 6.8: Imputation performed by drImpute and velocity inference of scVelo are integrated as pre-processing steps of GRISLI. As in Figure 6.7, from left to right, each box corresponds to one combination of imputed data matrix. It is evident that the imputation of both data matrices affect the AUROC score by increasing it by more than a hundredth. Imputation methods can definitely improve the quality of data and, as a consequence, the methods used later perform better.

is done with a more accurate method, such as scVelo, the performance of GRISLI is increased and the reconstructed GRN is more accurate. Even if the AUROC increases by just a hundredth, this is a significant development that should be studied even more thoroughly.

The objective of this thesis can be considered achieved, as GRISLI method has been improved in two ways. On the one hand, the data can be pre-processed through the utilisation of an imputation method, which is a way to reduce the percentage of dropout events and, consequently, to avoid computational problems due to the sparsity of the expression matrices. On the other hand, the exploitation of a performing and well-designed velocity inference method can be crucial in order to obtain a solid estimate of the real RNA velocity.

7 Conclusion

The objective of this work was to improve the performance of the GRISLI algorithm through imputation of data and a better inference of the RNA velocity.

In the first place, the fundamental definitions of the biological concepts such as RNA velocity and gene regulatory networks are introduced. Some state-of-the-art methods for their estimation are explored in detail, revealing their main assumptions and the differences among them. Then, a more technical discussion is made about the simulation of single cell data and its imputation. The algorithms that have been used to simulate and impute data are analysed in depth.

In Chapter 6, some preliminary analyses of the performance of GRISLI are presented, as well as an evaluation about the computational cost in terms of time that is required. Afterwards, the results of different experiments are shown and discussed. First of all, it is observed that most of the existing RNA velocity inference methods have a bias towards negative velocity. In particular, negative values are found to be associated with genes with null expression. This fact is unexpected, since a gene that is not expressed cannot be more switched off than that. Thereafter, an analysis about the reasons for this negative velocity bias is completed. Exploiting the simulation methods, it was possible to generate various datasets which are affected by different amounts of dropout. It is confirmed that the abundance of null values within single cell data is the main cause of the tendency to infer many negative values for the RNA velocity. At this point, two imputation methods are used to diminish the effect of the null values in gene expression data.

The combination of imputed data and the availability of advanced RNA velocity inference methods, such as scVelo, is shown to be effective for the improvement of GRISLI performance. Indeed, in the final part of this work, this modified version of GRISLI is applied on a real scRNA-seq dataset and an increase in terms of AUROC score is achieved.

References

[1] B. Hwang, J. H. Lee, and D. Bang, "Single-cell RNA sequencing technologies and bioinformatics pipelines," *Experimental & Molecular Medicine*, vol. 50, pp. 1–14, 2018 Aug 01.

[2] G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, J. Fan, L. E. Borm, Z. Liu, D. van Bruggen, J. Guo, X. He, R. Barker, E. Sundström, G. Castelo-Branco, P. Cramer, I. Adameyko, S. Linnarsson, and P. V. Kharchenko, "RNA velocity of single cells." *Nature*, vol. 560, pp. 494–498, 2018 Aug 01.

[3] V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis, "Generalizing RNA velocity to transient cell states through dynamical modeling." *Nature Biotechnology*, vol. 38, pp. 1408–1414, 2020 Dec 01.

[4] V. A. Huynh-Thu and G. Sanguinetti, "Gene regulatory network inference: an introductory survey," *Methods in Molecular Biology*, vol. 1883, p. 1—23, 2019 Jun 06.

[5] S. Chen and J. C. Mar, "Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data." *BMC Bioinformatics*, vol. 19, pp. 232–, 2018 Jun 19.

[6] P. C. Aubin-Frankowski and J. P. Vert, "Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference." *Bioinformatics*, vol. 36, no. 18, pp. 4774–4780, 2020 Sep 15.

[7] B. Treutlein, Q. Y. Lee, J. G. Camp, M. Mall, W. Koh, S. A. M. Shariati, S. Sim, N. F. Neff, J. M. Skotheim, M. Wernig, and S. R. Quake, "Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq." *Nature*, vol. 534, no. 18, pp. 391–395, 2016 June 01.

[8] L.-F. Chu, N. Leng, J. Zhang, Z. Hou, D. Mamott, D. T. Vereide, J. Choi, C. Kendziorski, R. Stewart, and J. A. Thomson, "Single-cell RNA-seq reveals novel reg-

ulators of human embryonic stem cell differentiation to definitive endoderm." *Genome Biology*, vol. 17, no. 18, pp. 173–193, 2016 Aug 17.

[9] W. V. Li and J. J. Li, "An accurate and robust imputation method scImpute for single-cell RNA-seq data," *Nature Communications*, vol. 9, p. 997, 2018 Mar 08.

[10] W. Gong, I.-Y. Kwak, P. Pota, N. Koyano-Nakagawa, and D. J. Garry, "DrImpute: imputing dropout events in single cell RNA sequencing data." *BMC Bioinformatics*, vol. 19, pp. 220–230, 2018 June 08.

[11] H. Hochgerner, A. Zeisel, P. Lönnerberg, and S. Linnarsson, "Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing," *Nature Neuroscience*, vol. 21, pp. 290–299, 2018 Feb 01.

[12] V. Bergen, R. A. Soldatov, P. V. Kharchenko, and F. J. Theis, "RNA velocity—current challenges and future perspectives." *Molecular Systems Biology*, vol. 17, no. 8, p. e10282, 2021.

[13] A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. M. Murali, "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data," *Nature Methods*, vol. 17, pp. 147–154, 2020 Feb 01.

[14] H. Matsumoto, H. Kiryu, C. Furusawa, M. S. H. Ko, S. B. H. Ko, N. Gouda, T. Hayashi, and I. Nikaido, "Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation," *Bioinformatics*, vol. 33, no. 15, pp. 2314–2321, 2017 Apr 01.

[15] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature Biotechnology*, vol. 32, pp. 381–386, 2014 Apr 01.

[16] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PloS one*, vol. 5, no. 9, p. e12776, 2010.

[17] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory

Networks in a Mammalian Cellular Context," *BMC Bioinformatics*, vol. 7, p. S7, 2006 Mar 20.

[18] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, "TIGRESS: Trustful inference of gene regulation using stability selection," *BMC Systems Biology*, vol. 6, p. 145, 2012 Nov 22.

[19] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society, Series B*, vol. 72, pp. 417–473, 2010.

[20] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg, "SC3: consensus clustering of single-cell RNA-seq data," *Nature Methods*, vol. 14, pp. 483–486, 2017 May 01.

[21] Y. Cao, P. Yang, and J. Y. H. Yang, "A benchmark study of simulation methods for single-cell RNA sequencing data," *Nature Communications*, vol. 12, p. 6911, 2021 Nov 25.

[22] P. Dibaeinia and S. Sinha, "SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks." *Cell System*, vol. 11, pp. 1–20, 2020 Sep 23.

[23] R. Cannoodt, W. Saelens, L. Deconinck, and Y. Saeys, "Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells," *Nature Communications*, vol. 12, p. 3942, 2021 Jun 24.

[24] X. Qiu, Y. Zhang, J. Martin-Rufino, C. Weng, S. Hosseinzadeh, D. Yang, A. Pogson, M. Hein, M. K. Hoi-Joseph, L. Wang, E. Grody, M. Shurtleff, R. Yuan, S. Xu, Y. Ma, J. Replogle, E. Lander, S. Darmanis, I. Bahar, V. Sankaran, J. Xing, and J. Weissman, "Mapping transcriptomic vector fields of single cells." *Cell*, vol. 185, pp. 690–711, 2022 Feb 17.

[25] C. Qiao and Y. Huang, "Representation learning of RNA velocity reveals robust cell transitions." *Proceedings of the National Academy of Sciences*, vol. 118, no. 49, 2021 Dec 07.

[26] M. Gao, C. Qiao, and Y. Huang, "UniTVelo: temporally unified RNA velocity reinforces single-cell trajectory inference," *bioRxiv*, 2022 Apr 27.

Acknowledgments

I am extremely grateful to my supervisor, professor Gabriele Sales, who guided me during my internship and the writing of this dissertation. I would like to extend my sincere thanks to his research group as well, who have always been of help to me whenever I needed them.

I could not have undertaken this journey without my friends and my roommates. They have supported me over all these years and our memories together will remain indelible.

Words cannot express my gratitude to my parents, who have been understanding with me on every occasion and without whom I would never have come this far.

Special thanks are reserved to my brother, who has always shared moments of happiness and lightheartedness with me, especially in the most difficult times.

Padova, 21st September 2022 Greta