DECIPHERING CELLULAR HETEROGENEITY BY SINGLE-CELL TRANSCRIPTOME ANALYSIS

Lam-Ha Ly

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Berlin, April 2022

Erstgutachter: **Prof. Dr. Martin Vingron**
Zweitgutachter: **Prof. Dr. Hanspeter Herzel**

Tag der Disputation: 11. Juli 2022

“There is always light if only we’re brave enough to see it, if only we are brave enough to be it.”"

— Amanda Gorman (The Hill We Climb)

To my mum Dr. Thuy-Le Phan

PREFACE

PUBLICATION & CONTRIBUTIONS

For several studies in this thesis work was undertaken by many collaborators that are listed in the following.

- In Chapter 4.1, the first study resulted from a collaboration between the lab of Udo Reichl and Martin Vingron. The experiments with the virus infected cells, including cell preparation and isolation was performed by Sascha Kupke. Stefan Börno performed the single-cell RNA-seq on the cells. Stefan Haas investigated and supervised the computational analyses. The results were published as a shared first authorship between Kupke, Ly and Börno in *Viruses* (Kupke et al., 2020).
- In Chapter 5, Virginie Stanislas provided literature research information and performed some initial analyses on the experimental dataset.
- In Chapter 6, the presented study was published in *Patterns*, (Ly et al., 2021).

ACKNOWLEDGEMENTS

Doing a PhD and writing the thesis is demanding. Pursuing a PhD with a kid is even more challenging. However, completing a PhD with a kid during a pandemic is just insane. Therefore, I have to thank a bunch of people making it possible to reach where I am now.

First, and foremost I thank my supervisor Martin Vingron for giving me the opportunity and the necessary supportive environment to work and grow both my research and personal skills in his lab. It has been a fruitful mentorship based on mutual respect and encouragement and I am beyond grateful for calling him my PhD supervisor.

I thank Stefan Haas for his co-supervision especially in the first time period of my PhD providing me constructive feedback, his caring assistance throughout my PhD and for proof-reading my doctoral thesis. I appreciate Hanspeter Herzel for being part of my thesis advisory committee and my second reviewer of my dissertation.

I would also like to acknowledge the facilities at the Max Planck Institute such as the IT helpdesk and the sequencing facility. Special thanks goes to Paul Menzel and Thomas Kreitler who solved every IT-related issue reliably and very quickly as well as Stefan Börno for his collaborative work and expertise about processing single-cell RNA-seq data.

I am extremely grateful for being part of two graduate schools, the Computational Systems Biology (CSB) school and the International Max Planck Research School for Biology and Computation (IMPRS-BAC). Both gave me the opportunities to meet fellow students that I could exchange ideas and advises with, but also to attend interesting conferences and workshops to develop my scientific skills. At this point I would like to express my sincere gratitude to the amazing PhD coordinators Mary Louise Grossman and Cordelia Arndt-Sullivan from the CSB graduate school as well as Kirsten Kelleher and Anne-Dominique Gindrat from the IMPRS-BAC graduate school. Everyone of them has provided and organized useful scientific workshops and events. More importantly, they have assisted me in childcare related issues and always have had an open-ear for my worries and difficulties. Thank you, Kirsten, for also proof-reading parts of my thesis.

I would also like to acknowledge the whole Vingron department for their scientific input and helpful discussions. Special thanks goes to Prabhav Kalaghatgi and Virginie Stanislas for providing me feedback about the dissertation. As my office mates, Virginie and Maryam Ghareghani, have always giving me a good reason to come to the office and sweeten my office life with little snacks, laughter and chats. I very much appreciate my former colleague Lisa Barros de Andrade e Sousa for her supportive and encouraging friendship and my colleague Hossein Moeinzadeh for his valuable input and conversations. During the pandemic he and his family shared childcare with us and established a supportive and safe environment to work and to accommodate the kids.

I would like to extend my sincere gratitude to many people outside of academia. I very much appreciate my parents in law, Christine and Manfred Hartmann, for taking care of their grand child while I was extending my working hours in the office. Special thanks go to Lisa-Marie Göppert, Khanh-Ly Nguyen, my brother Hong-Linh Ly, my sisters Lam-Tuyen Ly and Lam-Thanh Ly for their emotional support and their empowering words throughout my academic career. A sincere and deep thank you to my mum, Thuy-Le Phan, and dad, Truong-Van Ly, for your guidance, your love and the basis you build my life on. Thank you for creating the stable backbone, a place I call home and the nutritious and delicious food every now and then. Words cannot express the grateful feeling to have people surrounding and cheering you in every step, in every goal, no matter how big or small it is.

Last, but not least I would like to express my deepest acknowledgment to my family. Thank you, Aaron Liem, for reminding me to set my priorities, flooding me with love and keep telling me I am the best human being of your world (for now). Thank you, Martin Hartmann, for always being there bearing my ups and downs and for putting aside your needs. Your mental support, your belief in my skills and your patience have been a major contributor to my PhD process.

ABSTRACT

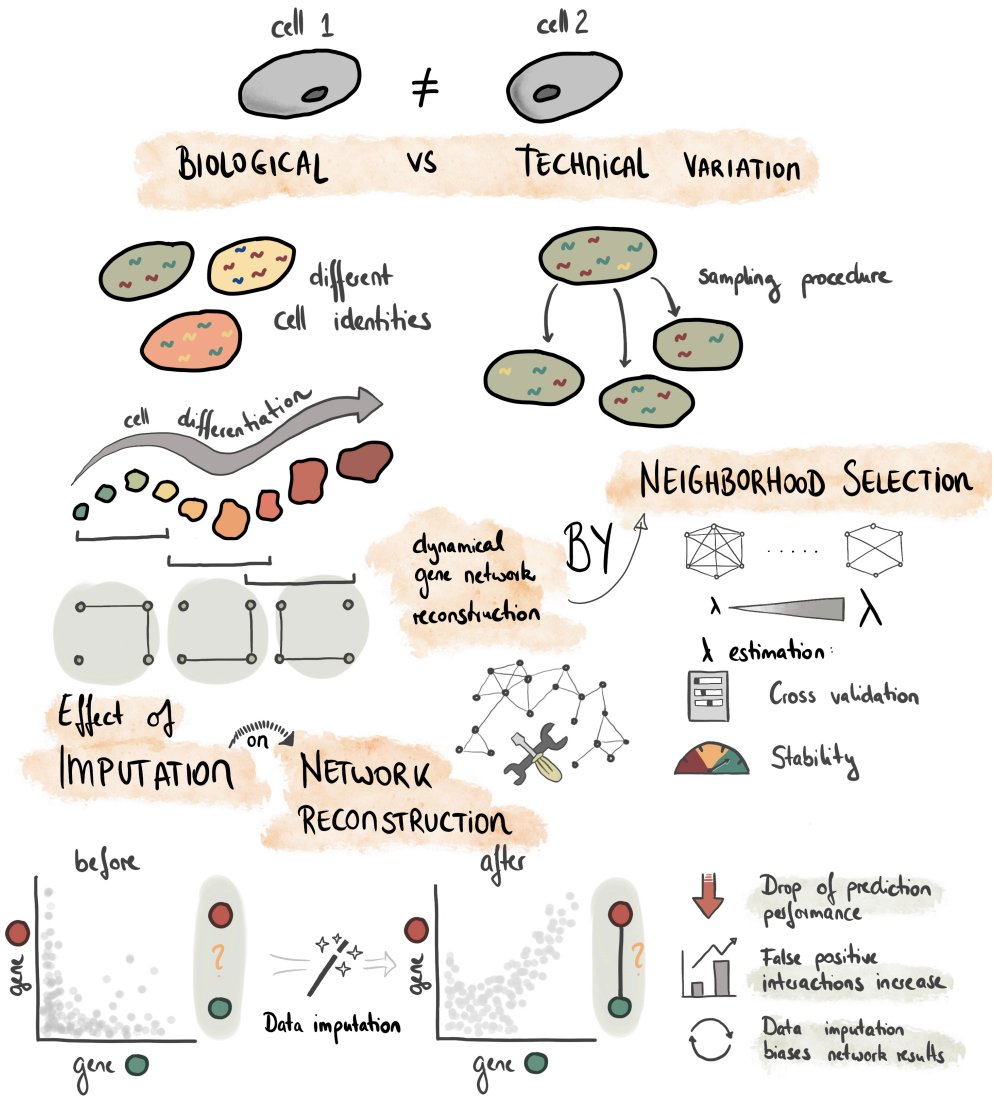
The advances of single-cell transcriptomics enable a plethora of new analytical approaches but also challenges. One of the main difficulties in single-cell RNA-seq data is to differentiate between the unwanted technical and desired biological variability across cells. In the first part of the thesis we show how to assess the technical variability in both experimental single-cell RNA-seq data and in a simulated framework that we established by downsampling single cells from bulk reference samples. In the course of this study we show how bulk RNA-seq samples can be integrated in a pre-computed topology trained on single cells. Furthermore we reveal biases in similarity measures between the derived samples that strongly depend on the gene detection rate of an experiment. In the second part we focus on how to interpret cellular variability by predicting regulatory interactions between genes in the context of network reconstruction. We implement the neighborhood selection method that uses a two-fold model selection criteria for parameter estimation. We apply the method on data generated *in silico* exhibiting different developmental trajectories commonly seen in single-cell biology. We dynamically infer time-dependent gene regulatory networks evolving through the course of temporally ordered trajectory and revealing active gene regulations in a particular time-frame. Furthermore, we systematically evaluate the effect of data imputation on gene regulatory network reconstruction. We observe an inflation of gene-gene correlations after data imputation that affects the predicted network structures and may decrease the performance of network reconstruction in general. Altogether this thesis provides insights about how to deal with the observed heterogeneity and how it can be used to infer regulatory associations between genes using single-cell transcriptome data.

ZUSAMMENFASSUNG

Mit dem Fortschritt in der Einzelzell-Transkriptombiologie ergeben sich viele Möglichkeiten verschiedenster und neuer analytischen Konzepte zur Beantwortung biologischer Fragestellungen. Zeitgleich birgt diese aber auch viele Herausforderungen. Eine der Hauptschwierigkeiten bei Einzelzell-Transkriptomdaten ist die Differenzierung zwischen der technischen und der biologischen Variabilität in Zellpopulationen. Ersteres ist artifiziell und sollte daher bei den Analysen außer Acht gelassen werden. Im ersten Teil der Arbeit wird gezeigt, wie die technische Variabilität sowohl in experimentellen Einzelzell-Transkriptomdaten als auch in einem etablierten simulierten Datenmodell gemessen werden kann. In dem simulierten Datenmodell werden Einzelzellen aus einer populationsbasierten Referenzprobe abgeleitet und auf verschiedene statistische Eigenschaften untersucht. Im Verlauf dieser Studie wird aufgezeigt, wie klassische, populationsbasierte Transkriptomproben in eine vorberechnete, auf Einzelzellen trainierte Topologie integriert werden können. Darüber hinaus werden verschiedene Ähnlichkeitsmaße zwischen den abgeleiteten Proben betrachtet und Verzerrungen beobachtet, die stark von der Gendetektionsrate eines Experiments abhängen. Der Fokus des zweiten Teils der Arbeit liegt auf der Interpretation der zellulären Variabilität durch die Vorhersage von regulatorischen Interaktionen zwischen Genen im Kontext der Netzwerkrekonstruktion. Die implementierte Methode *neighborhood selection* verwendet ein zweifaches Auswahlkriterium, um einen geeigneten Parameter für die Netzwerkrekonstruktion zu schätzen. Deren Anwendung findet auf Daten statt, die *in silico* generiert wurden und unterschiedliche, in der Einzellzellbiologie üblich vorkommende Zelldifferenzierungsverläufe aufweisen.

Unter Hinzuziehung der Daten werden dynamische, genregulatorische Netzwerke abgeleitet, die sich im Laufe einer zeitlich geordneten Trajektorie entwickeln und aktive Genregulationen in einem bestimmten Zeitrahmen offenbaren. Darüber hinaus liefert die Arbeit eine systematische Evaluierung über die Auswirkungen der Datenimputation auf die Rekonstruktion genregulatorischer Netzwerke. Es wird eine inflationäre Zunahme der Gen-Gen-Korrelationswerte nach der Datenimputation beobachtet, die sich auf die vorhergesagten Netzwerkstrukturen auswirkt und die Prognosefähigkeit der Netzwerkrekonstruktion im Allgemeinen mindern kann. Insgesamt liefert diese Arbeit Erkenntnisse darüber, wie mit der beobachteten Heterogenität in Einzelldaten umzugehen ist und wie sie genutzt werden kann, um aus Einzelzell-Transkriptomdaten zuverlässiger auf Assoziationen zwischen Genen schließen zu können.

Deciphering Cellular Heterogeneity by Single-Cell Transcriptome Analysis



CONTENTS

1 INTRODUCTION

Cells are the smallest units of life and represent a complex biological system. Every multi-cellular organism evolves from a single cell that differentiates into tissues and develop complex organs. In 1957, Conrad Waddington introduced the landscape of cell fate commitment which illustrates cell differentiation (Fig. 1.1). It represents a ball rolling down a landscape consisting of hills and valleys. Along the ball's path it reaches intermediate branching points at which the ball can either take the left or right direction. Once the ball reaches an end point it represents the differentiated cell state. The idea beneath the landscape is that gene regulatory processes happen controlling the cell fate's decision. One of the key points in systems biology is to comprehend the factors regulating cell differentiation that form complex multi-cellular tissues.
With the advances of single-cell transcriptome data it is possible to "follow" a cell differentiation path by arranging the cells in a temporal order along the path. This procedure enables to identify key factors that determine the cell fate's decision or the cellular identity. These key factors are often called transcription factors regulating the transition to another cell type or maintaining a cell state. They are of particular interest as they control their target gene's activity level. Once deleted or modified, transcription factors can cause the lost of cell identity inducing dysfunction or diseases. Identification of transcription factors causing diseases can be used to develop medical treatment to target towards them.

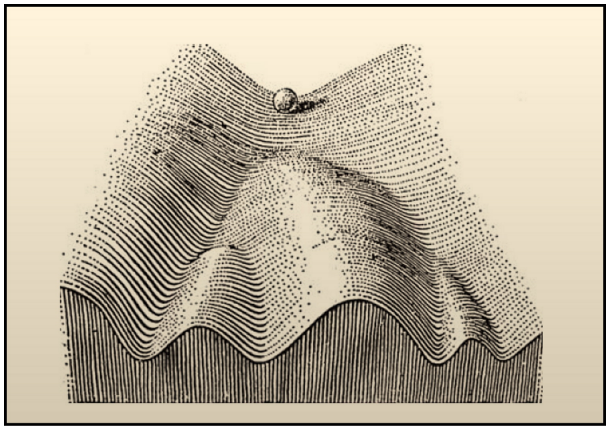


Figure 1.1: The Waddington landscape. A model to illustrate cell fate commitment. A ball goes through a path within the landscape depicting the differentiation path. Along the path, there are branching points at which the ball is either taking the left or right path. Once the ball reaches the end point, it is in a differentiated state.

In this thesis we use single-cell transcriptome data to unravel cellular heterogeneity in order to study gene regulation. One of the first goal of the thesis was to have some hands-on experience on single-cell transcriptome data. The aim here was to investigate Influenza A virus infected cells and explain the cell-to-cell variability in the viral production rate across cells. However, through the course of this study it became apparent that the experimental data composition was challenging to deal with. It prevented us from disentangling the observed heterogeneity into the technical variability (noise) and the biological variability (signal) within the cells that we are interested in. Having faced this difficulties, we moved on with developing a simulation framework

where we could study the technical variability in a controlled environment. In the course of the simulation study we examined also features such as the integration between bulk and single-cell samples and the investigation of similarity measures between a reference bulk sample and its derived simulated single cell.

In the second part of the thesis we aim to investigate gene regulation by reconstructing gene networks from single-cell transcriptome data. After providing a short mathematical introduction and summary about state-of-the-art algorithms, we apply a method based on the concept of *neighborhood selection* to infer regulatory networks. We evaluated the proof-of-concept on simulated data and examined gene regulatory networks evolving through a temporal ordered trajectory. We tested the method on experimental data but only saw weak performances compared to other network reconstruction algorithms. We attempted to apply data imputation that smooths the data and predicts values where no information is available, prior to network reconstruction. However, the neighborhood selection did not profit from imputation and resulted in a poorer performance than without data imputation. We asked whether this is a method-specific artifact or a systematic problem of imputation in general. To test this, we study systematically the interplay between imputation and network reconstruction in multiple state-of-the-art algorithms across different experimental datasets. We show that data imputation can enhance gene-gene correlations significantly influencing network structures such that the network reconstruction performance decreases upon imputation.

Thesis outline

Following this Chapter, a detailed biological introduction is given in Chapter 2. It provides some background information relevant to this thesis in biology and gives a broad overview about technologies performing transcriptome-wide experiments.

Chapter 3 provides an introduction about computational challenges and steps relevant for a data analysis on single-cell transcriptome data. It is separated into a preprocessing and a downstream analysis part.

Chapter 4 deals with assessing the variability in single-cell data. The first part covers the challenges that we faced when we attempted to distinguish between the technical and the biological variability in an experimental dataset. The second part investigates the variability and other technical features in a simulated framework.

Chapter 5 investigates transcriptome data with respect to gene regulation in the context of gene networks. We use a mathematical concept based on neighborhood selection to infer gene regulatory networks in a static and dynamical way using temporally ordered data. We apply the method on several datasets, both simulated and experimental data.

Chapter 6 examines the combination between data imputation prior to gene regulatory network reconstruction and evaluates the predicted network models. Lastly, the conclusions and future insights of this thesis are summarized and discussed in Chapter 7.

2 BIOLOGICAL BACKGROUND

This chapter introduces the fundamentals in molecular biology relevant to this thesis. It serves as a baseline in order to understand the biological mechanisms throughout the thesis. Moreover it provides an overview of technologies for performing transcriptome profiling in bulk populations as well as single cells. The information in the first part (Chapter 2.1 – 2.2.1) covering the basics in molecular biology until transcriptome profiling in bulk populations is mainly extracted from the textbook *The Molecular biology of the Cell* by Alberts et al., 2014 if not stated otherwise.

2.1 CENTRAL DOGMA OF MOLECULAR BIOLOGY

Each cell contains genetic material that preserves the hereditary information about its organism, the ***genome***. It stores the information as a double-stranded helix known as the ***DNA*** (Fig. 2.1A). Each strand is of a sequence of four different nucleotides consisting of a sugar molecule, a phosphate group as well as one of the four nucleobases: Adenosine (A), Thymine (T), Guanine (G) and Cytosine (C). Two nucleic strands are able to bind to each other via hydrogen bonds following a complementary base pair (bp) pattern discovered by Watson and Crick in 1956: adenosine–thymine and cytosine–guanine.
In order to pass the genetic information to the next generation, the DNA needs to be duplicated. This process is called ***replication***. In eukaryotes, replication happens during mitosis in the course of the cell cycle: The double-stranded DNA gets unwound such that a part of the DNA is split up into two single strands serving as templates. Now, by complementing the two single strands from free nucleotides in the cell new DNA gets synthesized and distributed among two cells.

The ***expression*** of information stored in the DNA to a functional molecular unit in the cell is summarized in the central dogma in molecular biology (Fig. 2.1B) and covers two processes: First, the ***transcription*** from the DNA to an intermediate nucleic single-stranded molecule, the ***RNA***, and secondly, the ***translation*** passing information from RNA to the synthesis of the functional gene products, the ***proteins***.

Transcription. During transcription a segment of DNA (***gene***) is read and copied into a processable RNA (often messenger RNA (***mRNA***)). Here, proteins, so called transcription factors (TFs), along with enzyme, the RNA polymerase, bind to the a regulatory region, called promoter, at the 5' UTR (untranslated region) (see Figure 2.1C). Transcription initiation starts and the RNA polymerase synthesizes by recruiting the matching nucleotides the complementary strand. In eukaryotes a 5' cap is added to the 5' end in order to ensure that the mRNA does not get degraded. The transcription process remains until the RNA polymerase reaches the transcription stop site at the 3' UTR. A poly(A) tail, a stretch of adenine bases, is added to the 3'UTR and the RNA polymerase releases from the DNA. The precursor mRNA (Fig. 2.1C) is synthesized and is

post-transcriptionally modified by a process called ***splicing*** in which the non-coding regions (introns) are removed while coding regions (exons) remain. Now, the mature mRNA is formed and serves as a template for protein synthesis.

Translation. The last process, translation, is the decoding of a sequence written in a 4-letter nuleotide alphabet to a sequence written in a 20-letter amino acids alphabet, the proteins. Here, ribonucleoproteins called ribosomes read the mRNA in a moving triplet base pair window, the codon. Each codon encodes an amino acid that the transfer RNA (tRNA) attaches to the synthesized amino acid chain. The end products, the proteins, fulfill the majority of a cell's function: They act as enzymes for catalytic reactions, regulators during gene expression, cell surface markers and many more.

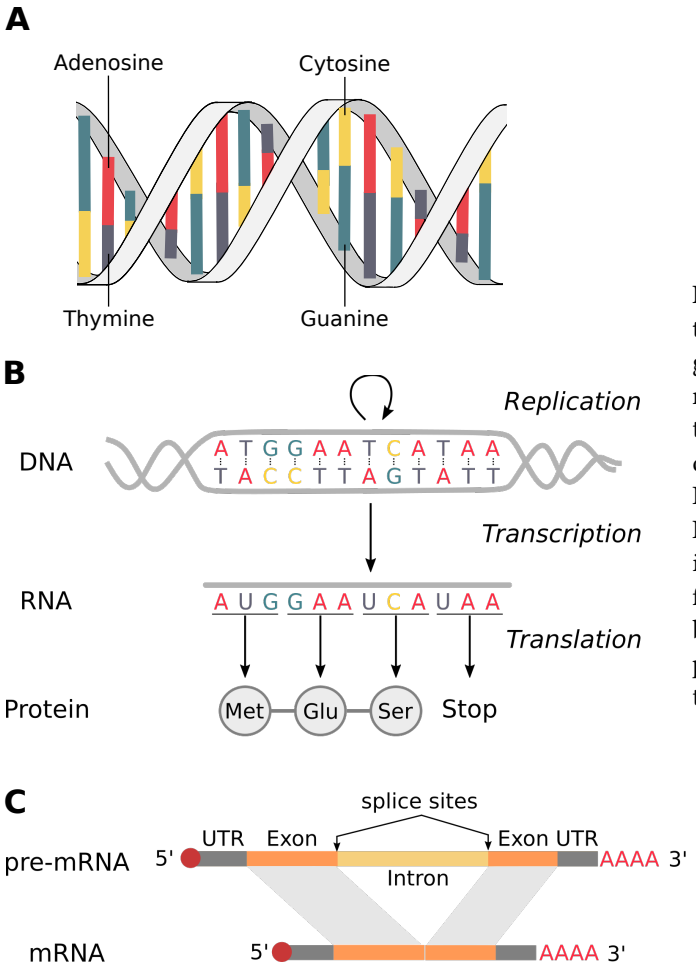


Figure 2.1: The genome and how the cell reads the genome. (A) The genome is a double helix with four nucleic bases: adenosine, thymine, cytosine and guanine. **(B)** The central dogma of molecular biology. From DNA over RNA to proteins. **(C)** Post-transcriptional RNA processing in eukaryotes. pre-mRNA is copied from a DNA segment and processed by 5' capping, RNA splicing and 3' polyadenylation to generate the mature mRNA.

2.2 TRANSCRIPTOME PROFILING

Each multi-cellular organism derives from one cell, the zygote, and differentiates into many cell lineages forming tissues. Each cell contains the same underlying DNA. However, the cells vary

across tissues which in turn consist of of many cell types. The difference of these cells is shaped by the amount of expressed genes, the transcripts. The total amount of transcripts, the ***transcriptome***, comprises all RNA molecules including protein-coding and non-coding transcripts. The transcriptome gives rise about a cell's identity and thus, there is great interest in profiling the transcriptome. It allows us to get more insights into gene regulation and to better understand and characterize cell types.

One way to profile the transcriptome is to measure the abundance of synthesized mRNAs at a current time point of a biological sample. Hence, we obtain a snapshot of the gene expression activity within a sample. Nowadays, a sample can be a cell population, tissue or a single cell. The next sections cover a historic overview of technologies used for profiling the transcriptome from hybridization to sequencing techniques in bulk. Then, we move to sequencing technologies in single cells and cover different technologies for measuring gene expression in individual cells.

2.2.1 From hybridization to sequencing technologies

For the first time the simultaneous profiling of thousands of genes was possible with the development of ***DNA microarrays*** in the 1990s (Lockhart et al., 1996; Schena et al., 1995). DNA microarrays are a chip-based technology that compares the gene expression level of two samples. It makes use of a process called hybridization in which thousands of short DNA fragments (oligonucleotides) are used. Attached to the chip, these oligonucleotides serve as a probe for the sample's mRNA. Once extracted from the biological sample, the mRNAs are converted to complementary DNA (cDNA) and labeled with fluorescence. Comparing two different samples, each sample gets either a red or green fluorescent marker. Now the labeled cDNAs can hybridize to the oligonucleotides attached to the microarray. Here, the chips are designed such that an array spans the whole transcriptome of the biological sample. Thus, the sequence of the transcripts needs to be known beforehand such that it can be captured. Otherwise, it cannot be hybridized and is missed for further analysis. A microscope reads out the changes of gene expression values as an image. The colors can be read as the following: A red dot represents a higher expression level for the red-dyed sample, and vice versa for a green dot. A yellow dot means an approximately equal expression level in both samples. The intensity of the dot represents the amount of captured transcript.
In summary, DNA microarrays provide a relatively inexpensive way to quantitatively measure gene expression changes between two samples. However, they have some important drawbacks: Firstly, the sequences of mRNA samples need to be known beforehand and secondly, the read out is a relative measurement between the two samples and does not represent absolute measures of the sample's gene expression.
As prerequisite in DNA microarrays, one needs to know the sequence of the mRNA samples. Hence, determining the nucleotide sequence has been of great interest since long before the development of hybridization technology. In 1977, Frederick Sanger developed a method called ***Sanger sequencing*** to determine the sequence of a DNA fragment (Fig. 2.3). Here, a short oligonucleotide, called primer, is used, along with a DNA polymerase. A mixture of normal deoxynucleotides (dNTPs) is added, as well as chain-terminating nucleotides (dideoxynucleotides (ddNTPs)) which are labeled with four different fluorescence markers. Firstly, the primer hybridizes to the DNA fragment.

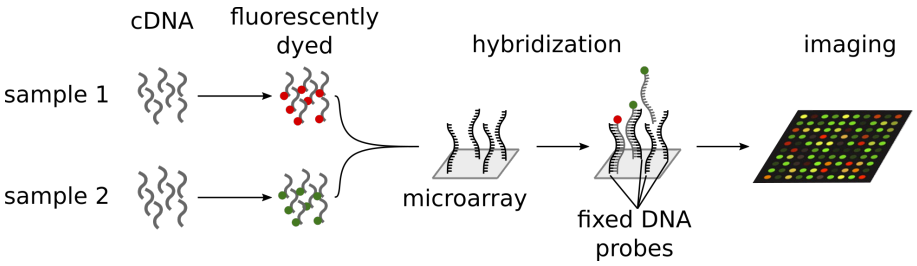


Figure 2.2: DNA microarray technology. DNA microarray to capture gene expression changes in two samples. mRNAs are extracted from two samples and converted to cDNA. cDNAs from sample 1 and 2 are labeled with red and green fluorescence markers, respectively. After mixing the cDNAs, the cDNAs hybridize to the oligonucleotides attached to the microarray. A microscope reads out the signals. Reprinted from Alberts et al., 2014.

Secondly, the DNA polymerase extends the primer sequence by complementary base pairing with four normal nucleotides. Occasionally, a ddNTP gets paired such that the copied sequence is terminated. This results in a set of DNA fragments with different lengths terminating with dyed ddNTPs. Thirdly, the set of DNA fragments are loaded onto a long glass capillary. Using electrophoresis the fragments are sorted by size. Then, a detector records the color of each dye labeled DNA fragment. Finally, a software reads the data and assembles the sequence.

With this technology it was possible to determine the genome of several species. However, with Sanger sequencing it was only possible to determine the sequence of limited length (up to 1000 bp). For this reason, the genome of interest was fragmented into random sizes using ***shotgun sequencing***. Then, each DNA fragment, called ***read***, is sequenced separately and afterwards assembled to a longer DNA sequence. This ***assembly*** is an approximate reconstruction of the whole genome. In the early 2000s, the human genome was first sequenced representing a major breakthrough (Collins et al., 2003). However, due to the low throughput, the process to sequence the human genome was very tedious: This project lasted more than a decade and required many scientists collaborating worldwide.

With the next generation, sequencing became high-throughput. The ***second-generation sequencing*** lowered the cost and speed dramatically by massively parallel sequencing. As an example, the company called *Illumina* provides a sequencing platform. Here, in brief, the experimental workflow is divided into four main parts covering library preparation, cluster amplification, sequencing and data analysis (Bentley et al., 2008).
Firstly, during library preparation the genome is randomly fragmented into segments of similar size. In case of transcriptome sequencing, the RNA is converted into cDNA beforehand. At both ends of the DNA fragments, adapters ligate and the whole sequence gets amplified by a process called polymerase chain reaction (PCR). As a result, multiple copies of the sequence are generated.
Secondly, during cluster amplification, the library is loaded into a flow cell and again gets amplified massively using bridge amplification. This step ensures that the same copies of the original DNA fragment get amplified in proximal space, generating clusters of DNA fragments.
Thirdly, these clusters are sequenced simultaneously. Here, Illumina uses specifically designed dNTPs that reversibly attach to the DNA template strand. The dNTPs are labeled with four

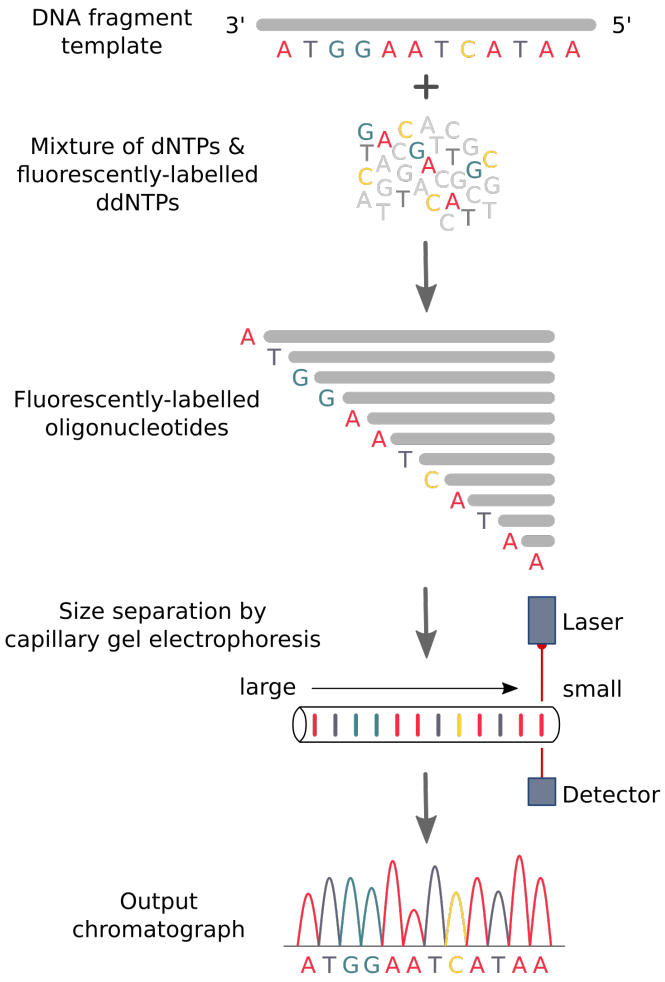


Figure 2.3: Automated Sanger sequencing. A DNA fragment of interest serves as a template to determine its sequence. A mixture of dNTPs and fluorescently-labeled ddNTPs are used to sequence short oligonucleotides. Occasionally ddNTPs are incorporated in the elongation phase such that oligonucleotides are generated with with varying lengths. The oligonucleotides are size separated by a capillary gel electrophoresis. A laser excites the fluorescent label and a detector catches the light emitted providing the label for each terminal ddNTP. The result output is a chromatograph showing fluorescent peaks for each ddNTP which can then be assembled to determine the sequence of interest.

different dyes emitting light once incorporated to the strand. A detector captures the color of the four nucleotides added to each template strand in each cluster. Thus, simultaneous tracking of the colors for each cluster gives rise to the sequenced read on a base-by-base accuracy.
Finally, data analysis is performed and all the sequenced reads are put together computationally by either aligning or mapping them to a reference genome or by assembling the reads if there is no reference given. Having RNA transcripts sequenced, the reads can now be quantified to get the transcriptional profile of the sample.

Sequencing the transcriptome of cell populations is known as bulk RNA sequencing, or in short ***bulk RNA-seq*** (Mortazavi et al., 2008). In contrast to DNA microarrays, bulk RNA-seq does not need to know the sequences of the samples beforehand. This tool allows sequencing of the set of polyadenylated mRNA molecules within a cell population on a base-by-base resolution. Hence, besides differential expression analysis it also offers opportunities to compare the samples in a sequence specific manner, e.g. by studying alternative splicing events. Thus, since its discovery, bulk RNA-seq provides a wide range of applications especially in comparative studies in biological research and medicine, e.g. from evolutionary studies comparing the same tissues across different

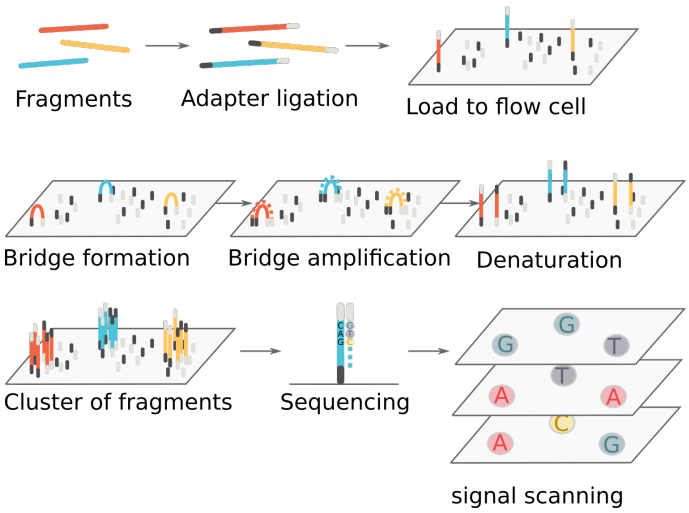


Figure 2.4: Illumina sequencing. A DNA fragment of interest serves as a template to determine its sequence. A mixture of dNTPs and fluorescently-labeled ddNTPs are used to sequence short oligonucleotides. Occasionally, ddNTPs are incorporated in the elongation phase such that oligonucleotides are generated with with varying lengths. The oligonucleotides are size separated by a capillary gel electrophoresis. A laser excites the fluorescent label and a detector catches the light emitted providing the label for each terminal ddNTP. The resulting output is a chromatograph showing fluorescent peaks for each ddNTP which can then be assembled to determine the sequence of interest.

species (Brawand et al., 2011) studying differences between healthy and disease samples or comparing cell populations before and after a treatment.

2.2.2 From bulk to single cell resolution

Sequencing the transcriptome can give valuable insights into a cell population. However, in order to perform bulk RNA-seq experiments, one needs cell populations consisting of thousands to millions of cells. Here, the polyadenylated amount of mRNAs of the whole population of cells derived from tissues or cell line samples are extracted and purified together before being sequenced. As a result, the average gene expression level across cells is measured. Although this gives us useful insights into comparative studies as stated above, this technology is nonetheless insufficient if we want to analyze complex or heterogeneous tissues such as brain or cancerous tissues. The tissues consist of many different cell types which are partly unknown. As a consequence, taking the average signal per gene across all cells within the samples might mask biologically relevant variation between cells.

Another limitation of bulk RNA-seq experiments is the lack of large cell populations in some research fields, e.g. when studying cell lineages upon cell differentiation. As an example, during early embryogenesis, a single fertilized zygote divides into more and more pluripotent cells, reaching intermediate cell states until finally forming a whole tissue or organism. This cell state change into a more specialized cell type is a popular but complex research field and little is known about the regulatory processes controlling cell fate decisions. Thus, especially with regard to cell differentiation there is a strong motivation to study the dynamic processes in gene regulation

with a single-cell resolution.

Overall, there has been great interest in studying the transcriptional profile by assaying gene expression at a single-cell level for multiple reasons. One aim was to identify previously unknown and possibly rare cell sub-populations in heterogeneous tissues, but also to study their cellular frequency composition (Stegle et al., 2015). Another aim was to identify (new) marker genes specifically expressed in certain sub-populations but also to investigate the dynamic gene expression pattern across cell differentiation and to ultimately understand the regulatory relationships between genes (Kolodziejczyk et al., 2015b).

2.2.3 From ten to hundreds of thousands of cells

Within a decade the number of individual cells being sequenced in one experiment grew exponentially (Svensson et al., 2018). It started with just a few dedicated cells to hundreds of thousands and now up to even millions of cells. Generally, in order to generate single-cell gene expression data from a biological sample one has to follow an experimental procedure including four main steps (Kolodziejczyk et al., 2015b; Luecken et al., 2019):

1. single-cell dissociation
2. single-cell isolation
3. library construction and
4. sequencing

Starting with the biological sample of interest, the first step requires *dissociation* of the sample (for example a tissue) into individual cells. Therefore, a single-cell suspension is added in order to digest the extracellular matrix of the sample loosening cell-cell junctions (Reichard et al., 2019). The next step, the *isolation* of single cells depends on the use of the experimental technology. The following section summarizes key technologies highlighting major differences occurring during single-cell isolation and library construction. Furthermore, they represent major jumps regarding the scalability in the number of sequenced single cells.

In 2009 **Tang et. al.** succeeded in sequencing the transcriptome of a single cell for the first time (Tang et al., 2009). Here, the experimental protocol was adapted to the low genetic starting material by improving the amplification of cDNA from single cells in an unbiased way. However, the isolation of the single cell by manually selecting a single cell under the microscope prior to the sequencing step was tedious. For this reason the following studies sequenced only a couple of cells mainly focusing on early embryonic developments and later on cancer cells (Ramsköld et al., 2012; Tang et al., 2010, 2011).

Sample multiplexing. Initial parallel sequencing of almost 100 single cells was performed by Islam et. al. using a multiplexing method called single-cell tagged reverse transcription (STRT)-seq (Islam et al., 2011). Here, ***multiplexing*** in the context of sequencing refers to a procedure where the samples are pooled together to be processed and sequenced simultaneously (Wong et al.,

2013). More specifically in STRT-seq, each cell is loaded into a well of a 96-well plate. During library preparation, a barcode is introduced and used as a tag for each reverse transcribed cDNA for each cell allowing the assignment of the transcripts to the cell. After pooling the cells with their corresponding cDNAs as one single mixture, this mRNA mixture gets sequenced. However, instead of sequencing the full length of each mRNA transcripts, only the 5' end of the transcripts gets sequenced. Now, using the barcodes each transcript can be assigned to its respective cell. Regarding cell isolation another improvement was introduced with this technology: Instead of manually isolating the cells into tubes, a semi-automatic device was developed to select the single cells, loading them into the wells. As an alternative, researchers also used fluorescence-activated cell sorting (FACS) in order to select and to load the cells.

Integrated microfluidic chip. With the development of microfluidic chips (C1) by the company *Fluidigm*, automatic cell capturing became possible, simplifying the cell isolation step (Xin et al., 2016). Here, the cells are loaded onto a chip. By sequentially flowing through the chip, the cells are captured in chambers. Within these chambers biomolecular reactions can be performed including reverse transcription and PCR amplification in order to prepare the sequencing library. After that, the library can be extracted and loaded into the 96-well plates before sequencing. An advantage of using this C1 system is that it is compatible with multiple single cell RNA-seq protocols. Suffering from the manual selection before, the mRNA-seq protocol (SMART-seq) used in Ramsköld et al., 2012, the combination of the C1 system now allows an automatic way of cell capturing of several hundreds of cells with a full-length transcript sequencing procedure.

Droplet-based microfluidics. Technologies based on droplet emulsions enabled the next big jump into several thousands of sequenced cells. InDrop (Klein et al., 2015) and Drop-seq (Macosko et al., 2015) protocols and also the commercially available 10x Genomics platform use barcoded beads encapsulating the cells in a microfluidic device. Commonly, each bead has a cellular barcode assigning the cell's membership and biochemical reagents for library preparation. After reverse transcription and PCR amplification taking place in each bead for each cell individually, the beads get lysed. Then, all captured transcripts are pooled and purified before sequencing. Similar to STRT-seq, these protocols do not capture the full-length transcript but only the 3' end of each transcript.

Combinatorial *in-situ* barcoding. Most recently, another concept was developed called single-cell combinatorial-indexing RNA-seq (sci-RNA-seq) and is based on a combinatorial way of multiple indexing cycles. It allows for profiling up to hundreds of thousands or even millions of cells. Instead of isolating single cells, the technique randomly distributes a small pool of cells into microwell plates. Each well with its corresponding pool of cells gets a unique barcode that is integrated *in situ* to the mRNA of each cell. Then, all cells of the whole microwell plate are pooled and randomly redistributed into small pools across a plate repeatedly. As a second round, a well-specific barcode is integrated and all the cells are pooled again. This procedure can be repeated a third time. Using multiple cycles of random sorting and distributing the probability of two cells having the same sequence of wells and thus the same sequential barcode is arbitrarily low. In a final step, the genetic material is pooled and amplified to prepare the sequence library.

After library construction a *sequencing* machine produces the raw sequencing reads that need to be processed computationally. Similar to the bulk RNA-seq experiments, the generated reads need to be aligned to a reference genome and quantified. If a multiplex protocol has been used the data needs to be demultiplexed assigning the mRNA counts to its corresponding cell by the cellular barcode. This results in a final ***gene expression matrix*** which can be further analyzed.

The technologies summarized here represent available experimental procedures and platforms to profile the transcriptome on a single-cell level. Each of them have advantages and disadvantages which were assessed here (Svensson et al., 2017; Ziegenhain et al., 2017). The technologies do not only differ in the number of sequenced cells feasibly profiled in a single run or whether a full-length compared to a 5' or 3' RNA sequence tagging procedure is used but they also vary in their performances. There are differences regarding the sensitivity in capturing lowly expressed genes, the accuracy in the quantification of the actual gene expression level as well as the cost per cell. While designing a single-cell study, it is necessary to think thoroughly about the experimental setup as this influences the computational analysis to a great extent.

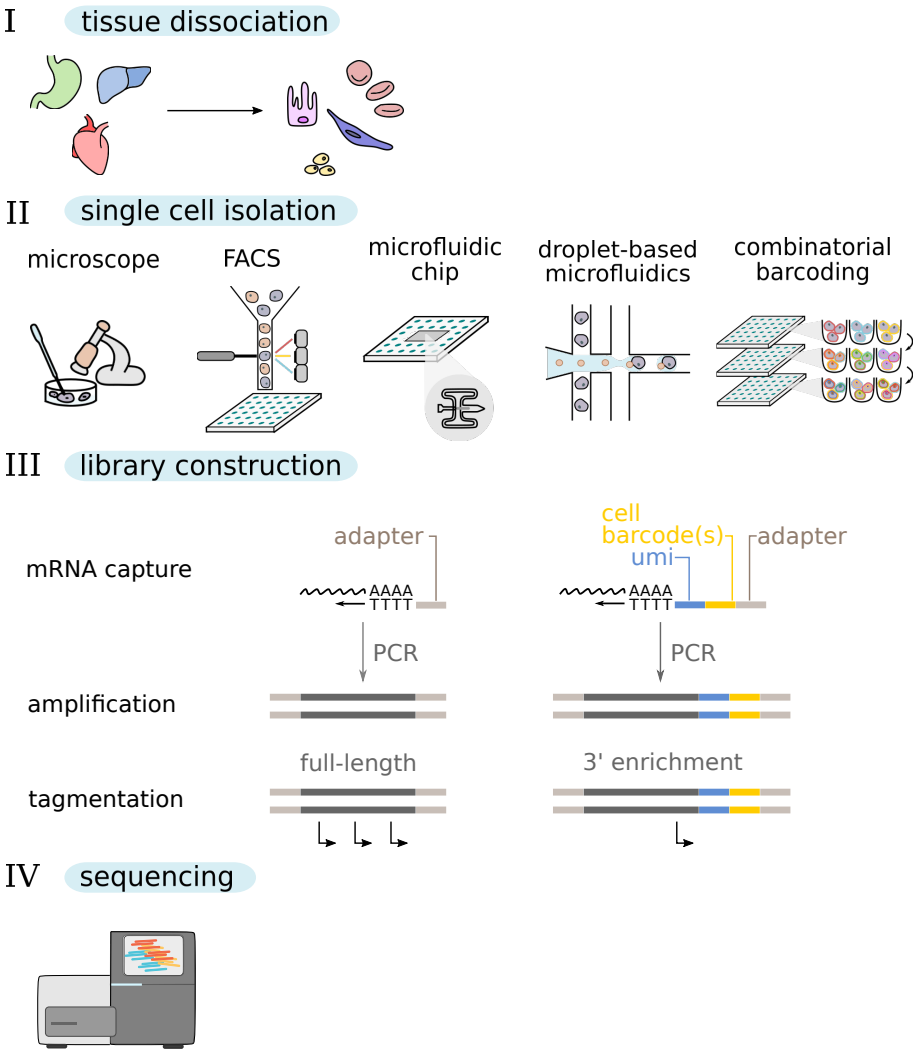


Figure 2.5: Single-cell experimental procedure. The four main steps of a single-cell experiment are (i) tissue dissociation, (ii) single-cell isolation, (iii) library construction and (iv) sequencing. (i) During tissue dissociation single cells are detached from their tissue environment and separated into individual cells. (ii) For single-cell isolation there exist many technologies including the manual selection with a microscope, FACS machine, microfluidic chip, droplet-based microfluidics and combinatorial barcoding. (iii) Library construction depends on the corresponding protocol. First cellular mRNA is captured by oligo dT primers (with or without umi/ cell barcodes) and amplified by polymerase chain reaction (PCR). Tagmentation produces the final sequencing library resulting either in full-length transcript or 3' or 5' enriched transcripts (not shown). (iv) The sequencing library is sent to the sequencing machine to generate the raw reads.

3 ANALYTICAL CHALLENGES AND STEPS IN SINGLE-CELL TRANSCRIPTOMICS

In recent years, the number of published single-cell studies has increased exponentially with growing numbers of sequenced single cells (Angerer et al., 2017; Svensson et al., 2020). Publicly available ***atlases*** derived from mouse tissues (Han et al., 2018), or whole model organisms such as zebrafish, frog or flatworm serve as reference maps and give rise to the identification of new cell types or the cellular composition of tissues or the organisms.
However, new insights from such studies can only be revealed through appropriate analysis and interpretation of the results. For this reason it is crucial to understand the "nature" of single-cell RNA-seq data in order to develop suitable computational tools which require modeling of the underlying statistical distribution of the data. Only when we have done this, and correctly interpret the results, can these tools assist in answering biological questions.

Dealing with single-cell data is not trivial. Researchers not only have to be aware of the different available experimental platforms but also the availability of the enormous number of computational tools. The rapid growth in single-cell transcriptomics lacks a consensus on how to analyze single-cell RNA-seq data computationally (Vieth et al., 2019). Depending on the hypothesis of a single-cell study, the question whether some analysis steps *can* or even *should* be used makes it difficult to agree on a consensus pipeline. Additionally, the consecutive combination of multiple analytical components may have a critical effect on the interpretation of the results.

For this reason, I first provide an overview about the sources of technical noise and biases occurring in single-cell data in order to understand the complexity and challenges associated with the data analysis. I summarize the individual computational components that (i) can be integrated into a ***preprocessing*** pipeline in Section 3.1 and (ii) can be applied onto the preprocessed data in order to explain the cellular heterogeneity in ***downstream analyses*** in Section 3.2. I mostly refer to the guidelines presented by Luecken et al., 2019: *Current best practices in single-cell RNA-seq analysis: a tutorial.*

3.1 PREPROCESSING SINGLE-CELL RNA-SEQ DATA

Collecting the sequencing data in a raw read count matrix, or a gene expression matrix, the data needs to undergo certain steps of *data cleaning* or also called ***data preprocessing***.
Preprocessing data in general is crucial and guarantees the comparability across the samples. It includes data cleaning, correction and normalization in order to ensure outlier and error removal as well as a standardized data format for the correct interpretation of the results in the data analysis. Many preprocessing methods have been developed for bulk RNA-seq data including normalization (Smyth et al., 2003) or data correction such as for batch effects (Johnson et al., 2007).

However, due to the particularities of single-cell transcriptome data that will be highlighted in the following sections, the previously available tools need to be adapted or re-developed.

3.1.1 Quality control

As every data analysis workflow begins, quality control represents the very first step before analyzing single-cell RNA-seq data. It ensures that low quality cells are filtered out and excluded from further downstream analyses as they might influence the analysis leading to misinterpretation of the data (Ilicic et al., 2016). There a different reasons for a sample being of low quality that we will now present. We will refer to the term *sample* instead of *cell* as we are uncertain if the sample as it occurs in our data matrix corresponds to an actual single cell.

During the automatic cell capturing process, the samples defined by the cellular barcode can be erroneous in two respects: The samples can either represent empty droplets/wells with no cell captured or even multiple cells (often called doublets). Here, different computational strategies allow for identifying empty samples or doublets. A simple procedure is to look at the library size and gene detection distribution and use a thresholding procedure in order to filter out samples with very low or high library sizes, or gene detection rates, respectively (Luecken et al., 2019). A more sophisticated alternative to identifying doublets is the application of an outlier detection algorithm (Ilicic et al., 2016).

Another reason for low quality cells are cells under stress due to damage received during the cell capturing process. These cells produce stress signals and might even undergo apoptosis misleading the interpretation of the biological results. Upon cell damage affecting the cell membrane, the cellular mRNA in the cytoplasm is abandoned while mitochondrial mRNA is conserved due to its compartmental membrane. Thus, a relatively low library size as well as low gene detection rates with a high amount of mitochondrial gene count per sample are indicative for damaged and thus low quality cells (Luecken et al., 2019). In addition, biological features such as an enrichment of expressed genes associated with apoptosis can also be used to identify low quality cells (Ilicic et al., 2016).

3.1.2 Discerning technical noise for normalization

Differences in the analysis between bulk RNA-seq data and single-cell RNA-seq data originate mainly in the modeling of the technical noise (or variability). Here, the technical noise is a result of the process of (i) capturing and (ii) amplifying mRNA transcripts within a single cell as well as (iii) varying sequencing depths across cells (Hicks et al., 2018; Kharchenko et al., 2014). When analyzing single-cell data it is crucial to take these factors into account.

mRNA capture efficiency. One of the most remarkable difference between bulk and single-cell RNA-seq data is the high number of zeroes dominating in the latter case. Oftentimes this phenomenon is called ***zero inflation*** and is attributable to the low amount of biological starting material. In general, there are two reasons for the presence of a zero count: (i) the gene was not expressed and thus no mRNA was synthesized referring to an actual biological zero count or

(ii) the gene was expressed, but the mRNA was not captured in the cell, possibly due to a low concentration, in which case we refer to a technical zero, often called ***dropout*** (Lun et al., 2016; Zhu et al., 2018). Technical zeroes are mainly due to the mRNA capture efficiency. It represents the sensitivity of an experimental protocol in capturing lowly abundant transcripts and differs across the technologies (Svensson et al., 2017).

Amplification bias. The second source of technical variation is due to the different amplification efficiencies of the captured sequences. By their structural properties such as length and GC content, the sequence composition introduces a bias during the amplification process (McDowell et al., 1998). In order to assess and to estimate the overall variability across the cells for each gene, a set of ***external spike-in RNAs*** has been introduced and can be used in single-cell experiments (Jiang et al., 2011). Here, a known and equal amount of different spike-in RNAs is added to each cell before library preparation. The set of external spike-in RNAs have different concentration levels mimicking the different expression levels of the cell's transcripts. Then, the cell's internal mRNA as well as the added external spike-in RNA are sequenced simultaneously. After sequencing and by quantifying the amount of spike-in RNAs we can compare the known concentration and the measured spike-in RNA abundance. The deviation between the input and the measured abundances provides an additional quality control but also a technical baseline that one can use for normalization (Brennecke et al., 2013; Lun et al., 2016). In fact, spike-in RNAs have been used in bulk experiments as well.

Furthermore, to account for the amplification bias, ***unique molecular identifiers (UMIs)*** are used in many experimental protocols (Sena et al., 2018). UMIs are short (6–10 bp) oligonucleotides providing barcodes that label each captured transcript in the cell. During library preparation the captured sequences are constructed in such a way that the sequences include (i) the reverse-transcribed mRNA, (ii) the cellular barcode labeling each cell and (iii) the UMI barcode. The constructed sequence is then amplified as a whole. By counting the *unique* pairs between cellular and UMI barcode it enables us to distinguish between amplified copies and actual biological copies of multiple mRNA transcripts of the same gene (Fig. 3.1).

Sequencing depth. Each sample has different number of sequenced reads, referred to as the sequencing depths. To establish comparability across the samples, it is necessary to normalize the samples' read count. Commonly used techniques for scaling across different sequencing depths is called ***counts per million reads (CPM)*** or ***transcripts per million reads (TPM)***. The latter normalization is used in full-length protocols considering the transcript sequence length. The assumption for both normalization techniques is that the initial amount of mRNA level in each cell is equal and that the sequencing depth varies due to a sampling procedure. This assumption holds true in bulk RNA-seq data. However, in single-cell RNA-seq it is uncertain how much mRNA content has initially existed within each cell. By profiling a tissue that consists of many different (partly unknown) cell types it is hard to predict which and how many cell types are present as the amount of mRNA varies across cell types. As sequencing of the transcripts happens in a multiplex procedure, the total amount of genetic material is pooled and simultaneously sequenced. Hence, cell-type specific factors and the variability in the mRNA capture efficiency contribute to the varying sequencing depths across cells. Here, spike-ins can be used to estimate the capture efficiency as well as the relative variability in cell size for UMI-based single-cell RNA-seq data

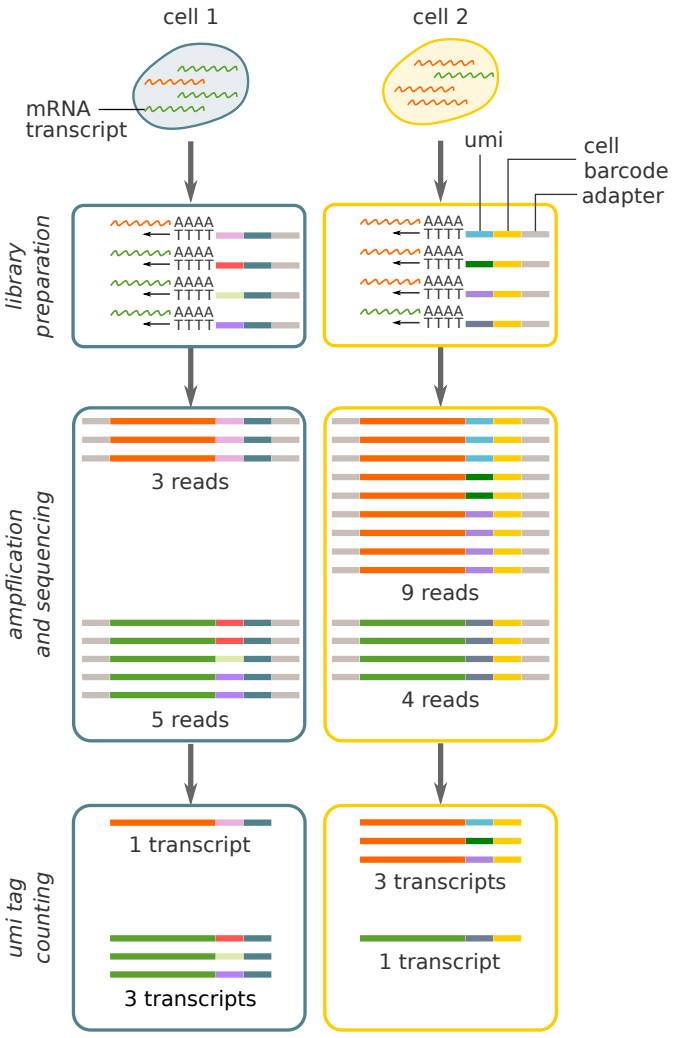


Figure 3.1: Correcting for amplification bias using UMIs. The polyadenylated mRNA molecules are captured in each cell. During library preparation UMI barcode, cell barcode and sequencing adapter are ligated to the captured mRNA molecule. The whole sequence is amplified and sequenced. The resulting reads consist of the sequencing adapters (on both ends), the cell barcode that labels the cell, the UMI barcode and the transcript sequence. During the amplification process, the quantitative amount of the reads derived from a transcript can deviate from the biologically available transcripts. This introduces a bias distorting the relative expression level across cells. By counting the unique pairs of transcripts and UMI barcode it is possible to eliminate that bias.

(Stegle et al., 2015, Fig. 2b,c).

In order to account for the unwanted technical noise including the above mentioned sources and the varying sequencing depths, computational methods have been developed for normalizing cell-to-cell variability accordingly with the option to use spike-ins and/or UMIs.

3.1.3 Correcting for systematic biases

In the former section we describe technical factors as a source of technical variability affecting single-cell RNA-seq data in an undesired way. Dependent on the biological question and hypothesis that researchers design, there can be also unwanted biological factors influencing gene expression levels that need to be taken into account. These factors are often referred as ***confounding factors***. They correspond to unobserved covariates which can blur the data signal and lead to misinterpretations when not correctly taken into account (Stegle et al., 2015). A simple example can be the presence of batch effects that also occur in bulk RNA-seq studies if experiments were performed on different days, laboratories or sequencing runs. Another case

occurs in studies examining e.g. differentiation processes. Usually these studies require a cell ordering according to the cell lineage from naïve to more differentiated cell states. Here, a possible confounding factor is introduced by the cell-cycle (Buettner et al., 2015). Cells in different stages of the cell-cycle show different gene expression patterns and thus might obscure the ordering of the cells according to the cell lineage. As a last example, another confounding effect occurs in studies investigating embryonic stem cell development. Oftentimes, samples of multiple early embryo tissues are selected before the development of sexual organs. Thus, the embryo's sex is unknown. However, sex-specific gene expression plays an important role already within the early days of embryonic development, thus causing differences in the gene expression signature. Integrating embryonic tissues from multiple embryos can show a diverging gene expression pattern caused by the different sexes.

While batch effects can be removed computationally by using the given annotation of the experiments, the two latter cases require preceding analysis in order to estimate either the cell-cycle state or the sex for each sequenced cell. Computational approaches allow the prediction of the cell state in order to account for unwanted biological variability (Buettner et al., 2015).

3.1.4 Distributional assumptions for normalization

For normalizing single-cell RNA-seq data many statistical models have been developed assuming different data distributions in order to account for the technical variability. Here, we present distributions that are commonly used to estimate the technical variability in the data.

Multinomial distribution. To model UMI-based counts in single-cell RNA-seq data oftentimes a multinomial distribution is hypothesized (Townes et al., 2019). The multinomial distribution is a generalization of the binomial distribution modeling counts with k categories and n number of independent trials. In case $k = 2$ (failure and success) the multinomial distribution refers to the binomial distribution.

For UMI read counts the multinomial distribution assumes the following: Let a single cell i contain t_i total amount of (unknown) mRNA transcripts and n_i be the total number of UMIs (measured) for the same cell. Note, that the UMI counts do not contain any PCR amplification biases (see Fig. 3.1). Due to the inefficient process of mRNA capturing the number of UMI counts is much lower than the actual number of mRNA transcripts within the cell ($n_i \ll t_i$) (Townes et al., 2019) . While UMI counts n_i can range between 1,000–10,000 the estimated number of mRNA transcripts is approximately 200,000 for a typical mammalian cell (Shapiro et al., 2013; Townes et al., 2019). Now, let x_{ij} denote the (unknown) number of mRNA transcript of a particular gene j in cell i, and y_{ij} the number of UMI counts of gene j in cell i. Also here applies $y_{ij} \ll x_{ij}$. Intuitively, genes with a higher number of available transcripts in the cell have a higher probability to be measured as a non-zero UMI count whereas genes with low abundances of the available transcripts have a lower probability to be measured and are likely to result in a zero count. Hence, the UMI counts y_{ij} follow a multinomial sampling procedure of the unknown actual counts x_{ij}.

Poisson distribution. In each each gene with abundance x_{ij} we either measure the gene's transcript or not in n number of trials. Here, the number of trials reflects the library depth in a sequencing experiment. Hence, the multinomial sampling can be approximated by the Poisson distribution. The probability mass function is given by:

$$P(X = l) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{3.1}$$

with l being the number of occurrences $l = 0, 1, 2, 3, ...$ and λ the intensity parameter. Here, we calculate the probability of a given gene to count its transcript l times with λ being proportional to the fraction of x_{ij} compared to all other gene counts in cell i. For Poisson distributed data there is a constant mean-variance relationship with:

$$\lambda = E(X) = Var(X) \tag{3.2}$$

The probability of observing a zero count is determined by:

$$P(X = 0) = e^{-\lambda} \tag{3.3}$$

Negative binomial distribution. As the Poisson distribution restricts the mean-variance relation to be constant, RNA-seq data is often modeled as a Gamma-Poisson mixture distribution or as an equivalent a negative binomial distribution (Anders et al., 2010). It allows the variance to grow in a quadratic relation to the mean and uses an additional parameter called *dispersion* to better model the variance-mean relation as we derive in the following. The negative binomial distribution is defined by two parameters: p as the probability of a single successful event (to observe a read count) and r as the number of failures occurring within the sequence of independent Bernoulli trials:

$$P(X = l|p, r) = \binom{l + r - 1}{l}(1 - p)^r(p)^l \tag{3.4}$$

Here, l denotes the number of successes. The negative binomial distribution has a mean μ (expected number of successes) computed as:

$$\mu = \frac{pr}{1 - p} \tag{3.5}$$

and variance σ^2 defined as:

$$\sigma^2 = \frac{pr}{(1 - p)^2} \tag{3.6}$$

Hence, solving Formula 3.5 to p with $p = \frac{\mu}{\mu + r}$ the variance σ^2 can also be expressed as

$$\sigma^2 = \mu\left(\frac{\mu}{r} + 1\right). \tag{3.7}$$

Note, the quadratic relation between the variance σ^2 and the mean μ in this formula leading to a quadratic increase of the σ^2 as a function of μ.

We can also express the probability function (3.4) subject to the parameters μ and r with:

$$P(X = l|\mu, r) = \binom{l + r - 1}{l} \left(\frac{r}{r + \mu}\right)^r \left(\frac{\mu}{\mu + r}\right)^l \tag{3.8}$$

Instead of r, one often uses the dispersion parameter ϕ which is the reciprocal of r. This gives us our final model with respect to its mean expression μ and dispersion parameter ϕ. Thus, the probability of observing a read count l is defined as:

$$P(X = l|\mu, \phi) = \binom{l + \phi^{-1} - 1}{l} \left(\frac{\phi^{-1}}{\mu + \phi^{-1}}\right)^{\phi^{-1}} \left(\frac{\mu}{\mu + \phi^{-1}}\right)^l \tag{3.9}$$

Similarly, the variance σ^2 in Formula (3.7) can be expressed by:

$$\sigma^2 = \mu\,(\mu\phi + 1)\,. \tag{3.10}$$

For observing "no read count with $l = 0$" we obtain the probability

$$P(X = 0|\mu, \phi) = \left(\frac{\phi^{-1}}{\mu + \phi^{-1}}\right)^{\phi^{-1}} \tag{3.11}$$

Zero-inflated negative binomial distribution. As single-cell RNA-seq data is dominated by large numbers of zeros researchers have modeled an additional distribution in order to account for the large numbers of zeros (Eraslan et al., 2019; Lopez et al., 2018; Risso et al., 2018). Besides of estimating the distribution of the count data the models add a probability of observing no read count in each gene by assuming a zero-inflated negative binomial distribution (Svensson, 2020). However, it is controversially debated if single-cell data needs to model an additional distribution estimating the 'zero-inflation' (Jiang et al., 2022).

3.1.5 Imputation

A series of computational methods originally developed for bulk RNA-seq fails to analyze single-cell RNA-seq data due to the distinct technical features and the high number of zero counts (Chen et al., 2018b). For this reason, imputation methods have been developed in order to counteract against the high number of zero counts by estimating values for zero counts referring to technical zeros and eventually smooth the data removing the noise. Thus, they are also called *expression recovery* or *denoising* methods. Imputation methods can be integrated within the preprocessing pipeline as an *option*. Whilst facilitating downstream analysis by improving gene-gene correlations, imputation methods however may also introduce false correlation signals (Andrews et al., 2018). In Chapter 7 we will further go into details about the effect of imputation methods on the reconstruction of gene regulatory networks.

3.1.6 Feature selection and dimensionality reduction

The human reference genome consists of about 20,000 protein-coding genes representing *features* in our data matrix. Hence, the raw count matrix is in a ~20,000 dimensional space. For many tools, this high-dimensionality is computationally expensive in terms of memory and run time. Feature selection provides a first step of reducing the dimensions by focusing on more *informative* genes. Here, we consider genes with high variability and high expression level across the cells as *informative*. We facilitate the analysis by removing genes with a constant and low gene expression level. However, the selection of the most informative genes is highly dependent on the steps prior to the selection step. Normalization accounting for the technical variability and data correction (as described in Sec. 3.1.2 and Sec. 3.1.3) can highly affect the selection of the most informative genes. Luecken et al., 2019 suggest using a subset of 1,000 – 5,000 most informative genes for further analysis.

After feature selection, another strategy to further reduce the dimension is to use an algorithmic approach. Here, the algorithms project the gene expression matrix into a low-dimensional space. Ideally, the projection decreases the noise level while retaining the main signal of interest. Here, we are interested in the signal that leads to the inherent structure of the data (Heimberg et al., 2016). Different algorithms exist in order to identify a lower dimensional space. However, the most commonly used approach is the ***principal component analysis (PCA)*** (Pearson, 1901). PCA is a linear approach that projects each data point (here cell) onto a lower-dimensional space by summarizing the data by their first principal components. These components explain the highest percentage of the variation of the data. Please note, that the variation does not have to be necessarily biologically meaningful (as referred in Section 3.1.2 and Section 3.1.3). Without explicitly accounting for the unwanted technical or biological variation it has been reported that the first or second principal component often correlates with a technical factor such as the fraction of genes expressed per cell (read count greater than 0) (Finak et al., 2015). Hence, principal components correlating with an unwanted factor should be discarded in the advancing procedure of data analysis. Finally, the lower-dimensional space provides a basis for the visualization of the data as well as other downstream analyses such as clustering and trajectory inference (Luecken et al., 2019).

3.1.7 Visualization

Visualizing the data is key in order to identify possible artifacts. We need to further reduce the dimensions in order to make the data "readable" for the human eye. For this reason, non-linear dimensionality reduction algorithms are commonly used. The two most prominently used algorithms are: t-distributed stochastic neighbor embedding (Maaten et al., 2008) and Uniform Approximation and Projection method (preprint: McInnes et al., 2018), or abbreviated ***t-SNE*** and ***UMAP***, respectively. While t-SNE captures well local similar structures in high-dimensional data, it suffers when representing global structures of the data. In other words, the visualization of the data is meaningful for data points within a cluster but less interpretable for data points across clusters (Kobak et al., 2019). In contrast to t-SNE, UMAP captures both local as well as global structures representing an accurate approximation of the underlying topology (Luecken et al.,

2019; Wolf et al., 2018). Besides for visualization purposes and similar to PCA, UMAP can also be applied to summarize high-dimensional data into a lower-dimensional space, but in a non-linear way. However, the components in the lower-dimensional space become less interpretable. Thus, PCA is preferred as a dimensionality reduction algorithm and UMAP as a visualization technique (Luecken et al., 2019). Another non-linear dimensionality reduction tool for visualization is based on ***diffusion maps*** (Haghverdi et al., 2015). Similar to a random walk approach it computes a Markovian transition probability matrix. The eigenvectors of the matrix represent the components that one can use for dimensionality reduction and visualization.

With the data visualization one can now explore and inspect the data by highlighting genes of interest (usually known ***marker genes***) by their gene expression value. This step leads to the first annotation of cell populations. In some cases, however, undesired data structures become visible. As an example, batch effects can be seen if the different batches (which need to be color-coded) separate within clusters. Oftentimes, the preprocessing steps need to get revised in order to proceed with the downstream analysis.

3.2 COMPUTATIONAL STEPS FOR DOWNSTREAM ANALYSIS

Single-cell transcriptomics has undergone an exponential growth not only in the number of published datasets (Angerer et al., 2017; Svensson et al., 2020) but also in the number of published tools analyzing single-cell RNA-seq data (`https://www.scrna-tools.org/`; Zappia et al., 2018). In November 2021, there exist 1115 tools which can be categorized into 30 different broad categories. This section outlines some analysis steps that are commonly used on single-cell transcriptome data. Basically, Luecken et al., 2019 distinguishes two types of downstream analyses: The cell-level and gene-level approaches.

3.2.1 Cell-level analysis

The cell-level approach describes the data on a global scale considering the cellular context. It includes computational analyses such as ***clustering*** and ***trajectory inference*** characterizing cellular structures.

Clustering allows for grouping cells with a similar gene expression pattern. By forming groups of cells, this approach tries to find substructures of cell populations. In the early days, the substructures were often declared *cell types* (Shalek et al., 2014; Zeisel et al., 2015). However, over time it has emerged as uncertain whether the speculative new cell types are truly biologically meaningful or rather an effect of the outcome of the computational analysis (Wagner et al., 2016). In particular, depending on the choice of clustering algorithm and the choice of parameters it may result into an arbitrary number of groups. For this reason, nowadays the common terminology describing the substructures is ***cell identities*** (Clevers et al., 2017; Luecken et al., 2019; Wagner

et al., 2016). With the help of known marker genes, the substructures can then be further annotated.

During a differentiation process, trajectory analysis allows the ordering of cells along a time path often known as ***pseudotime***. In a single-cell experiment, the cells are sequenced simultaneously and thus their individual transcriptome represents a snapshot within the differentiation process. However, the cells derive from a heterogeneous population spanning different stages of the dynamic processes during differentiation. Now, the trajectory analysis uses the dynamic processes inferring a temporal order of the cells. Moreover, it also allows for capturing differentiation paths diverging into multiple cellular end states or fates (bifurcation or multifurcation).

3.2.2 Gene-level analysis

Cell-level analysis compares and characterizes the cells with respect to their cellular context. Thus, it aims to *describe* the cellular heterogeneity. However, gene-level analysis dives deeper into the data and rather *uses* the cellular heterogeneity in order to investigate the molecular signals of the data. Thus, it aims to explain the question *why* do we see what we see. Here, we present two common approaches for substantiating the topology of the data: ***differential expression analysis*** and ***gene regulatory network inference***.

Even in bulk RNA-seq experiments differential gene expression analysis has been a common procedure for comparing two groups of experimental conditions in order to explain differences in gene expression. Bulk RNA-seq experiments are usually designed with two or three replicates for each condition. Hence, the bulk tools needed to account for gene variance in just a few samples. However, single-cell RNA-seq experiments do not have replicates *per se* but rather multiple cells with a similar gene expression profile (for example in a cluster). Cells derived from the same cluster differ across cells due to the technical noise as stated above. For this reason, methods designed to test differential gene expression in single-cell experiments model the cell-to-cell variability accordingly. In fact, it has been shown that differential expression analysis tools designed for bulk experiment perform as well as single-cell tools or even better if estimated gene weights have been introduced (Berge et al., 2018; Soneson et al., 2018). In any case, after testing for differential expression, p-values are assigned to each gene giving a measure of significance. However, even after correcting for multiple testing, it results into arbitrarily low p-values reaching values far below alpha significance level ($\alpha < 0.05$). Thus, statistically the majority of genes are considered significantly differentially expressed even though the biological meaningfulness is missing. This inflation of p-values is attributed to the computational design: Since usually clustering has been performed before testing for differentially expressed genes, the design forces the algorithm to reveal differences between the groups of cells. Here, one cluster is compared against all the remaining clusters. Hence, the identified differentially expressed genes are only specifically expressed within these pre-defined clusters. For this reason, there is a strong effort to develop methods testing for differential expression in a clustering-independent manner (Kim et al., 2020; Vandenbon et al., 2020). Nonetheless, the list of differentially expressed genes that might be filtered by the top ranks give us further insight into the biological data. Using a gene enrichment analysis an overrepresentation of a given gene set, for example pathways or other gene annotations using *Gene Ontology*, can be tested. It facilitates classification and further

characterization of the identified list of differentially expressed genes.

Another common gene-level analysis is the inference or reconstruction of gene regulatory networks (GRNs). This approach starts from the idea that genes do not operate independently from each other but rather in a complex network regulating each other's expression level dynamically. Uncovering the regulatory landscape from the expression data by predicting possible interactions between genes is one of the key aims to achieve in molecular biology. We will further go into more detail about the inference of GRNs in Chapter 5.

4 ASSESSING VARIABILITY IN SINGLE-CELL RNA-SEQ DATA

This chapter covers two scenarios of assessing variability in single-cell transcriptome data. The first scenario in Section 4.1 includes an experimental dataset published by Kupke et al., 2020. It is a study about Influenza A virus infected single cells and investigates the heterogeneity in the virus replication across cells. In this study our collaborators in the lab of Udo Reichl performed the experiments, while the single-cell RNA-seq experiments has been performed at the Max Planck Institute for Molecular Genetics by Stefan Börno. I processed the data and designed the computational workflow including preprocessing and analysis pipeline. The study was published in *Viruses* by Kupke et al., 2020. I will not go into detail of the published results but I will use the dataset in order to exemplify the computational challenges when facing experimental data.

The second scenario in Section 4.2 deals with a simulated framework. Unlike experimental datasets simulated single-cell transcriptome data provides a controlled environment in order to investigate the features of the data and its susceptibility to varying parameters without the influence of external factors. I implemented the simulation, thereby assessing the technical variability and investigated technical features such as the integration as well as similarity measures between bulk and single-cell derived samples.

4.1 A CASE STUDY IN INFLUENZA A VIRUS INFECTED CELLS

Influenza A viruses (IAV) cause respiratory diseases in humans - commonly known as the flu. A viral spread in the human population leads annually to seasonal epidemic outbreaks leading to high morbidity and mortality rates (Fauci, 2006). As pathogens, viruses infect a host cell hijacking the transcription machinery in order to replicate the viral genetic material as part of the viral life cycle (Fig. 4.1). This provides the blueprint for viral proteins to form new viruses, further denoted as viral particles, that are released through the host cell.

Viruses need a host cell in order to reproduce themselves. However, this reproduction rate varies from cell to cell. The rate ranges from just a few to up to multiple hundreds of released viral particles per cell (Heldt et al., 2015). In order to examine the reasons for the large cell-to-cell heterogeneity Kupke et al., 2020 designed the following experimental setup summarized in Figure 4.2. First, Influenza A viruses are injected into single cells using an initial viral amount of 8–10 multiplicity of infection (MOI). Then, after 12 hours of infection, the extracellular amount of released viral particles, the ***virus yield***, is assessed by the plaque forming unit (PFU) value. Small or large PFU values correspond to low or high productive cells with regard to virus yield, respectively. Next, low and high productive cells are selected and prepared for sequencing. For

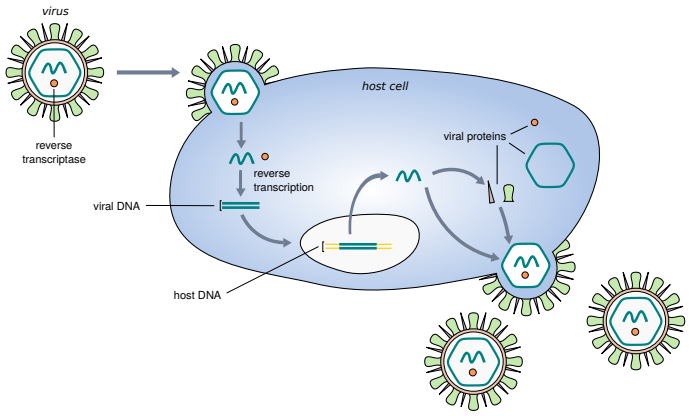


Figure 4.1: The viral life cycle. A virus infects a host cell and releases its genetic material into the cytoplasm. An enzyme, the viral reverse transcriptase, converts the viral RNA to a double-stranded DNA that gets integrated into the host's genome. The host transcription machinery is used in order to transcribe the viral genes to RNA to then synthesize viral proteins. The viral proteins are assembled to new viruses and are released from the host cell.

sequencing SMART-seq2 single-cell protocol is used to obtain full-length transcripts. Furthermore, external spike-in RNAs from the External RNA Control Consortium (ERCC) are added to the each IAV-infected single cell in order to assess the technical variability across the single-cell experiment.

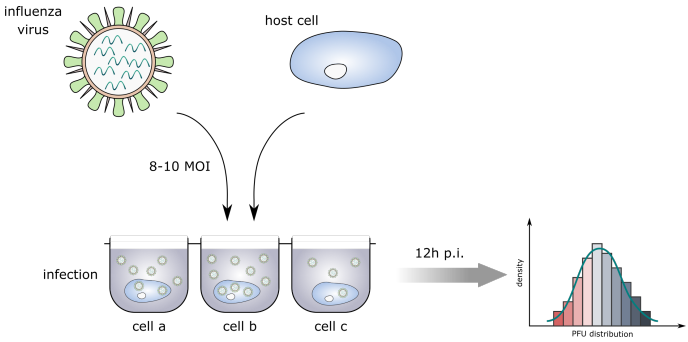


Figure 4.2: Experimental setup for Influenza A virus infection. Every host cell gets infected with the same initial viral amount (8–10 multiplicity of infection (MOI)) in a well. 12 hours post infection (p.i.) the extracellular amount of released viral particle is measured using plaque forming unit (PFU). Cells with the lowest and highest amount of PFU values are selected and prepared for sequencing.

In the following section, we demonstrate the complexity and difficulties along the analysis of single-cell transcriptome data faced in the context of this study. We focus on the preprocessing steps with an emphasis on the impact of the technical variability.

4.1.1 Removing systematic biases

Originally, the data consists of 96 single cells divided evenly into cells with a low and high virus yield. The sequencing data per cell profiled the host, pathogen as well as the spike-in RNA transcripts. Quality control was performed using a sequencing library size of >150,000 reads and an ERCC accuracy of >0.75. After quality control, 86 single cells remained with 45 low and 41 high productive cells.

For preprocessing we apply zinbwave (Risso et al., 2018) assuming a zero-inflated negative binomial distribution on the IAV-infected single-cell RNA-seq data. As a computational tool, zinbwave

can be used for normalization by modeling the data according to the statistical data distribution and by accounting for different library sizes per sample. Basically, the tool uses multiple regression models that can optionally incorporate unwanted *sample*-level covariates as well as unwanted *factor*-level (see Sec. 3.1.3). We plot the first two dimensions provided by zinbwave. Coloring the single-cells by the gene detection rate with a threshold of >1 TPM we observe a strong bias towards the gene detection rate (Fig. 4.3, left). Indeed, the Pearson's correlation coefficient between the first dimension and the gene detection rate is approximately -0.75. Therefore, we incorporate the gene detection rate as a sample covariate into the regression model thereby regressing out the unwanted technical factor. As Figure 4.3 (right) shows, the gene detection rate could successfully be accounted for. This is perceivable by the mix of single-cells with different gene detection rates. However, we do not observe a separation between high and low IAV-productive cells.

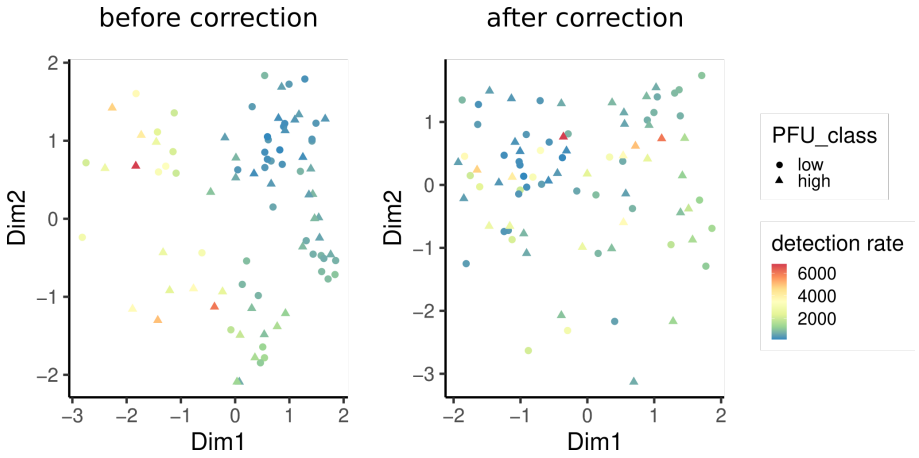


Figure 4.3: Zinbwave corrects for gene detection rate in IAV-infected single-cell RNA-seq data.
Low-dimensional representation of IAV-infected single-cell RNA-seq data using zinbwave before (left) and after (right) correcting the gene detection rate (color-coded) as a technical factor. High and low productive cells are coded by the shape.

Exploring the IAV-infected single-cell RNA-seq data we show how a technical factor (here the gene detection rate) can impact the result of a low-dimensional representation of the data. We successfully remove this effect using zinbwave. Nonetheless, the low-dimensional representation of the extended model does not show any separation between high and low productive cells. For this reason we suspect difficulties in examining differentially expressed host genes in order to explain the cell-to-cell heterogeneity in this dataset.

4.1.2 Technical vs biological variability

In general, the cell-to-cell variability in the single-cell RNA-seq data consists of two components: the technical and the biological variability. While the first is a matter of an unwanted technical effect attributed to noise in the single-cell experiment (see Section 3.1.2) the latter is the variability that one is interested in, assuming we already removed unwanted biological confounding factors. Assessing the technical variability, subsequently correcting for it and thus revealing the biological

variability is one of the key challenges in the preprocessing of single-cell transcriptome data.

The use of external spike-ins facilitates the assessment of the technical variability within an experiment. The ERCC set consists of 96 different transcripts with varying concentrations, mimicking the range of expression level of endogenous host transcripts. By comparing the known concentrations with the measured transcripts per million reads (TPM) expression values it is possible to quantify (i) the accuracy of the measurement per cell and (ii) the technical variability across the experiment. The former, the accuracy, is calculated by Pearson's correlation coefficient between the known and measured concentration level. The latter, the technical variability provides an estimate of the degree of noise level in the data. It is derived from the relationship between the squared coefficient of variation defined as $CV^2 = (\sigma/\mu)^2$ and the mean expression level μ for each gene across all cells. A constant amount of ERCC spike-ins with the same concentration profile has been added to each cell. Optimally, throughout the cells, the ERCC spike-in abundances should be consistently measured with little to no variation despite the strength of expression value. Hence, the coefficient of variation should be a constant line in relation to the expression value. However, due to the expected technical variability attributed to the sources of noise (mRNA capture efficiency and amplification bias), we expect the technical line to be a curve starting with high coefficient of variation in lowly expressed with a decline of variation in highly expressed ranges.

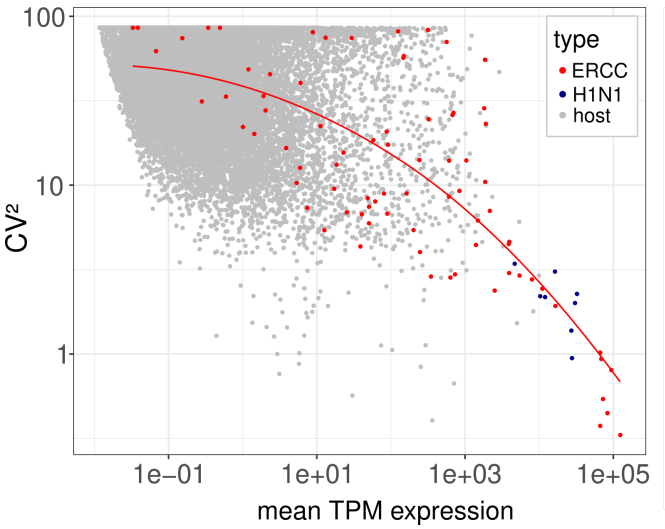


Figure 4.4: Assessing the technical variability using external spike-ins.
For each gene, the mean TPM expression value is plotted against the squared coefficient of variation for 86 single cells on a log-log scale. Host genes are color-coded in gray, viral (H1N1) genes in blue and ERCCs in red. The red fitted line represents the overall technical variability within the single-cell experiment.

Figure 4.4 shows the observed relation between the squared coefficient of variation and the mean TPM expression level per gene. As expected, lowly expressed genes have a high coefficient of variation that decline with a higher mean gene expression level. We use the ERCC spike-ins (colored in red) to fit a baseline representing the overall observed technical variability within the experiment. Notably, the spike-ins scatter with a high deviation around the estimated technical variability. This deviation of the individual ERCC transcripts give us a degree of how noisy the data is. ERCC transcripts with a low deviation from the fitted line imply an accurate estimate of the technical variability. This allows for identifying confidently biologically variable genes deviating significantly from the technical variabililty. However, if the ERCC transcripts have a high deviation from the fitted line, conclusions about biologically variable genes cannot be drawn

as they still fall in the range of technical noise.

Another observation that can be made from the plot is the high mean expression level of viral genes compared to the endogenous host genes. Indeed, comparing the transcriptional activity between the host and IAV genes, we observe a viral transcriptional activity that is significantly dominating the host activity level by three orders of magnitude (Fig. 4.5) This observation holds true for both, low and high IAV-productive cells. Interestingly, low productive cells show a significantly higher gene expression activity than high productive cells coinciding with a lower viral gene expression activity ($p < 0.005$ by Wilcoxon rank sum test).

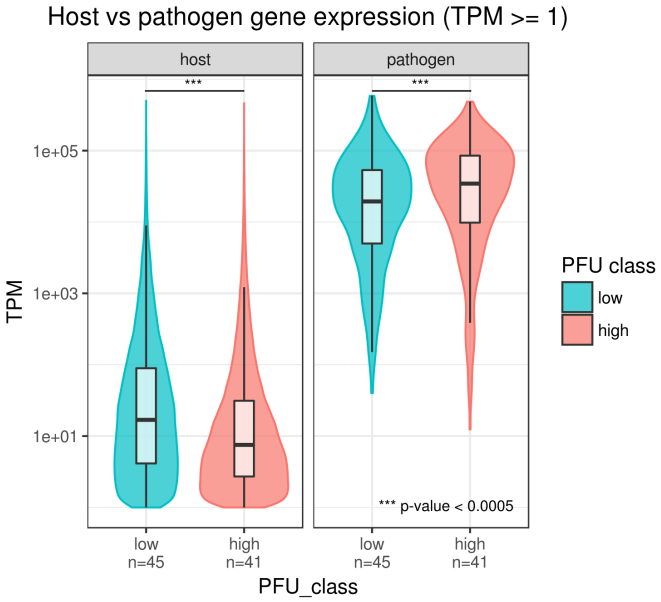


Figure 4.5: Transcriptional activity in host and pathogen between low and high productive cells. Expression values (TPM ≥ 1) for host and viral mRNAs are plotted on a log-scale for cells classified into low and high virus yields (low and high PFU). ***, $p < 0.0005$ by the Wilcoxon rank sum test Figure taken from Kupke et al., 2020.

In summary, the exploratory data analysis of IAV-infected single cells with regard to the host transcriptome reveals rather sobering results: Firstly, the low-dimensional representation of the data does not reveal any inherent separation between high and low productive cells. Secondly, the technical noise masks the biological variability to such an extent that statistically no conclusions on the host gene-level can be drawn. Finally, the majority of host genes have a much weaker signal than the viral genes that gets lost under the immanent technical variability.

4.1.3 Defective interfering particles affect the IAV replication

Although the noisy data quality prevents us from gaining any insights from the host expression profile, it still allows us to inspect the viral transcripts in more detail. In the matter of investigating the large cell-to-cell heterogeneity in the viral reproduction in the published study (Kupke et al., 2020), we discovered an enrichment of ***defective interfering particles (DIPs)*** in low productive cells. DIPs contain a long internal deletion in the viral genome segments leading to non-functional viral proteins. Upon co-infection these particles interfere with full-length viral segments in the

transcription process and thus contribute possibly to a decrease of released viral particles. Vividly speaking, DIPs act as pathogens for the viruses themselves. The study examines thoroughly the association between the co-occurrence of DIPs and the reproduction rate. However, as the detailed results go beyond the scope of the thesis we refer the reader to the actual study for further reading (Kupke et al., 2020).

4.1.4 Discussion

Aiming to explain the cell-to-cell heterogeneity in the viral reproduction we address several hypotheses/perspectives: The first perspective is that the variability may originate from the host cell. Conceivably, the host cell could have activated an immune response inhibiting the transcription of viral transcripts as previously shown (Russell et al., 2019; Timm et al., 2017). To test the hypothesis, the transcription of the cellular host genes needs to be analyzed and compared between low and high productive cells. In fact, we generally observe a significantly higher gene expression of host genes in low productive cells along with a lower expression of viral genes (Fig. 4.5). However, we could not relate this to an elevated immune response as the noise level originating from technical factors masks the biological signal leading to insufficient and noisy data. Additionally, the dominating viral gene expression saturates the host signal such that differential expression analysis led to no significant results. The second perspective to explain the cellular heterogeneity originates from the virus. Here, we found DIPs enriched in low productive cells. In the course of the cellular transcription process, DIPs interfere with viral transcripts and thus decrease the outcome of functional transcripts and in turn viruses.
The last perspective that could not be addressed due to the imbalanced transcriptional level of host and viral genes is the interaction between the two species. Although these different perspectives are listed independently they do not necessarily need to be treated as such. Arguably there exist different factors contributing to the wide range of viral replication in IAV-infected cells that still need to be elucidated.

4.2 INVESTIGATING SIMULATED SINGLE CELLS FROM BULK REFERENCES

4.2.1 Modeling single-cell count data

Motivation

As previously seen, real biological data can be challenging to analyze. Many unforeseen factors may influence the data introducing noise and biases that mask the biological signal.
Therefore, we were eager to set up a simulation framework to investigate the statistical characteristics of single-cell RNA-seq count data. The aim is to get a better understanding of the technical problems and pitfalls that govern single-cell RNA-seq data. A simulation framework provides a controlled environment to test our assumptions and a range of metrics by comparing the simulated data to a known ground truth. Here, the parameter settings enable us to regulate the simulation allowing us to investigate the direct consequences on the data composition.

In particular, in the early days of single-cell transcriptomics, bulk RNA-seq samples were often sequenced simultaneously to the single-cell samples in order to validate the performance of the experiment (Camp et al., 2015; Kolodziejczyk et al., 2015a; Shalek et al., 2014). The aggregation of all sequenced single cells allows us to compare their aggregated expression profile to the bulk reference. Ideally, the aggregated expression profile correlates well with the corresponding reference and confirms the proof of concept.

This section deals with the reverse scenario. In our simulation framework a bulk RNA-seq sample is used as a reference to generate single-cell-like data. The key point hereby is, that we are fully aware of the origin of the sample from which it is drawn enabling us to reveal different statistical properties between the reference and its derived simulated sample. Firstly, we can use the simulated data to approximate the bulk reference to the single cell population. To do so, we examine if bulk RNA-seq samples can be projected into the computed embedding of single cells thereby integrating bulk RNA-seq and single cell samples in the same topological map. Secondly, the data allows us to measure the technical variability by deriving multiple single cells from the identical bulk RNA-seq reference and assessing the variance across the cells. By this, we are able to investigate distributional assumptions of the technical variability. Lastly, we can study technical deviations between the single cells and the derived reference by measuring different similarity measures. More specifically, we can pose the question how much the simulated samples resemble the reference sample and which factors may influence the similarity to each other. Hence, the simulated framework provides an environment to investigate different statistical hypotheses and properties in order to gain more knowledge about single-cell RNA-seq data.

Data collection and processing

In order to study the above mentioned concept, we collect bulk transcriptome data from the Genotype-Tissue Expression (GTEx) portal (`https://gtexportal.org/`, Consortium, 2015). The database is a resource to study gene expression and regulation from non-diseased, post-mortal human tissues. In total, the GTEx dataset consists of approximately 11.6K samples, classified into

30 broad tissues, and 54 sub-tissues (Fig. 4.6). Note that Figure 4.6 visualizes samples color-coded by the broad tissues and can be topologically separated in the UMAP embedding referring to sub-tissues (not shown). The missing annotation of the sub-tissues has no impact on our further analysis. However, we find individual samples that fall into a different group of tissue samples and classified them as ***ambiguous***. For example, we can identify mismatched samples in the muscle cluster (top), the blood vessel cluster (bottom left) or even the skin cluster (bottom).

We are certainly aware that these ambiguous samples might be a consequence of the dimensionality reduction process. Here, the embedding displays the high-dimensional data onto a 2-dimensional representation. Of course, during this process, information may get lost and the data is "squeezed" into this low-dimensional space. Plotting the data in more dimensions would probably resolve the ambiguity of the samples. However, for the purpose of this simulation and the sake of simplicity we remove the "ambiguous" samples. This results in a final matrix of about 7,200 bulk RNA-seq samples and approximately 56,000 genes including protein-coding and non-coding genes.

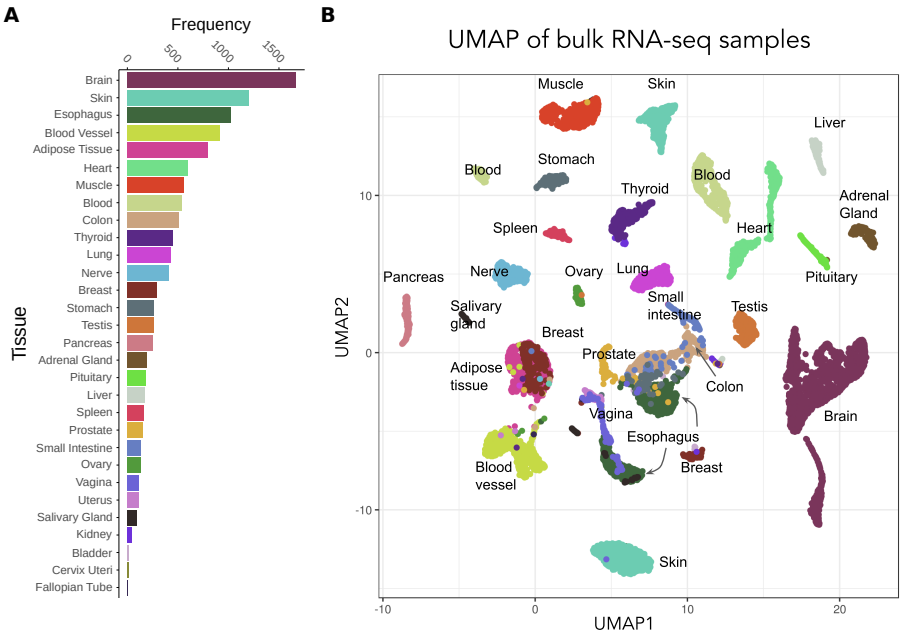


Figure 4.6: GTEx data comprises gene expression data for 30 different human tissues.
(A) Sample frequency of 30 different tissues. (B) Low-dimensional representation of GTEx bulk RNA-seq data color-coded by tissue annotation. Note, that tissues can be annotated further into sub-tissues and might be split apart in the embedding (for example samples derived from skin tissues).

Simulation scheme

Given any bulk RNA-seq sample providing a reference distribution, we simulate a single cell following a ***downsampling*** approach: We treat the bulk reference as a multinomial distribution

as it has been assumed previously (Baran et al., 2019) (see Section 3.1.4). We model the probability of occurrences with a m-sided die which is rolled s times. Here, m represents the number of genes and s represents the sequencing depth. Alternatively, we can illustrate the downsampling approach of a multinomial distribution by an urn experiment (Fig. 4.7). The number of balls m represents the number of the genes with their respective occurrences. We transform the occurrences into m subintervals. The length of the subintervals is equal to the corresponding ball occurrences within the urn. After transforming the intervals into probability intervals spanning the range between 0 and 1, we can generate random variables $X \sim \mathcal{U}(0,1)$, each time representing the drawing of a ball. We repeat the sampling procedure s times emulating the sequencing depth. Note that s is illustrated as a fixed number. However, in our computational realization, we draw the sequencing depth s from a normal distribution $\mathcal{N}(\mu, \mu/10)$ with mean μ and standard deviation $\mu/10$. Finally, we count the number of occurrences in each interval. A pseudo code for the downsampling procedure is provided (Algorithm 1).

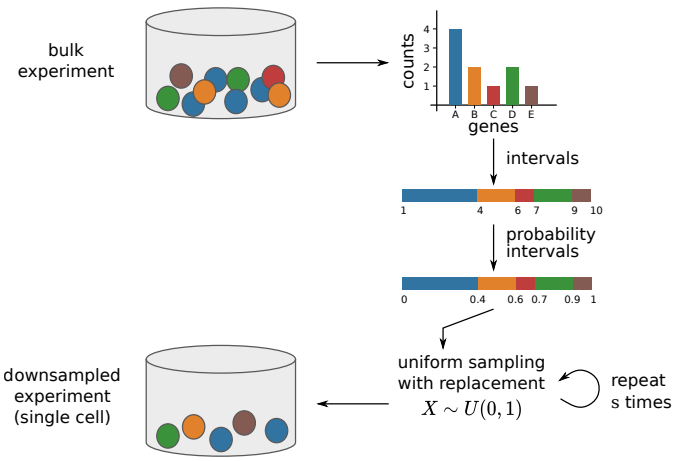


Figure 4.7: Simulation scheme. Bulk RNA-seq experiments illustrated as an urn experiment: Balls represent genes, the occurrence of a ball refers to its gene count. We derive the count numbers for each gene and divide them into intervals. The length of the interval reflects the count number. The intervals are turned into probabilities accumulating to 1. We sample from a uniform distribution between 0 and 1 s times as we would sample from the urn with the given probabilities and report the event. Finally, we count the number of occurrences for each gene and obtain the downsampled bulk experiment.

Algorithm 1: Simulation scheme

Input: count vector $x = (x_1, x_2, \ldots, x_m)$,
sequencing depth s

1) **init** $y_i = 0$, with $i = 1 \ldots m$
2) Compute probabilities $p_i = x_i / sum(x)$

repeat
  j = sample from 1 ... m with probabilities p;
  y[j] = y[j] + 1;
until *s times*;

return *downsampled count vector y*

4.2.2 Simulating bulk-derived single cells

Using the GTEx bulk RNA-seq samples and the presented simulation scheme, we simulate single cell samples. Overall, the simulated dataset comprises 1,600 single cells distributed over 16 different tissues (100 single cells for each tissue). We use two simulation scenarios with a sequencing depth of roughly 1,000 and 10,000 reads, respectively (Fig. 4.8). As Figure 4.8A shows, the simulated single cell can be downsampled from its reference bulk population resulting in a higher enrichment of lowly expressed genes. Furthermore, comparing the two simulation scenarios, we observe that the simulated library sizes scatter around 1,000 and 10,000 reads, respectively which is due to the normal distribution that the library sizes are drawn from (Fig. 4.8B). As expected, we see higher numbers of detected genes (read count ≥ 1) with a greater sequencing depth. After selecting the first 50 principal components, we displayed the simulated data using the first 2 dimensions the UMAP embedding. Here, the samples separate well. Hence, we established a simulation framework that generates bulk-derived single cells. We are now able to investigate further the simulated data according to our following statistical analyses.

4.2.3 Integrating single-cell and bulk RNA-seq samples

In this section, we want to investigate whether bulk RNA-seq and single-cell samples can be integrated into the same embedding. Computing a topological map with e.g. UMAP can be computationally very expensive especially if the datasets include a large number of sample sizes. In order to prevent computational re-calculations if new data is added to the analysis we aim to use the existing embedding to integrate the newly added data points to that embedding. In case bulk samples represent data points that shall be added to the embedding it allows us to characterize and annotate cell clusters by using the proximity of bulk samples to the closest single-cell clusters in an unsupervised way.

The concept that we develop to project samples into a pre-computed embedding is illustrated in Fig. 4.9. The idea is to train a model that computes an embedding for single cells. The trained model includes a dimension reduction using PCA. By choosing the top k principal components the model computes the UMAP embedding and projects the trained samples into the first two dimensions. Then, we use the bulk RNA-seq samples (or another set of single cells left out from the training procedure) and predict the PCA coordinates using the trained model. By using the predicted coordinates in the PCA space we further predict UMAP coordinates of the pre-trained UMAP embedding. By this we are able to use a pre-computed embedding in order to integrate different sources of data points into the same topological map.

We train our model using the simulated single-cell dataset with a library size of ~1,000 reads normalized to counts per million reads (CPM). The trained model includes a dimension reduction on the first 22 principal components and uses the reduced PCA space to compute the UMAP embedding on the first two dimensions. Instead of selecting a fixed number of principal components as it was done before we chose the first 22nd principal components based on a convergence criteria also known as the *elbow* approach (Thorndike, 1953). Using the trained model we predict

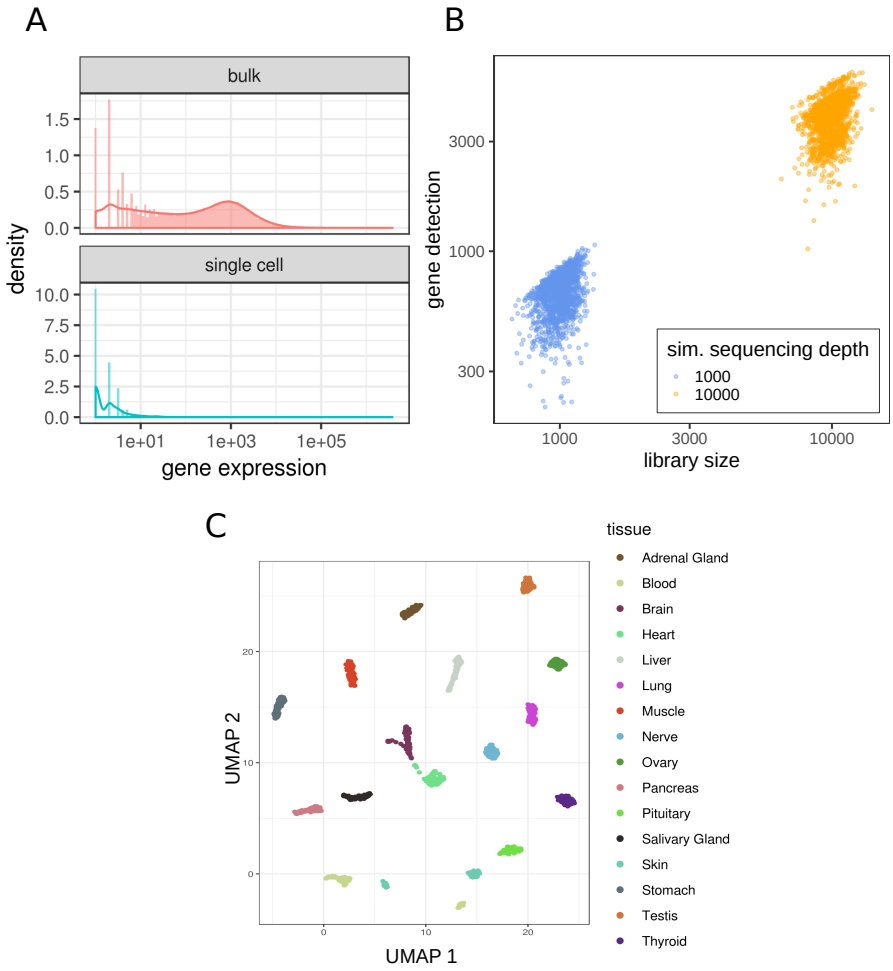


Figure 4.8: GTEx derived simulated single cells

(A) Gene expression distribution (log-scaled y-axis) between bulk and derived simulated single cell sample (exemplified). (B) Library size vs gene detection rate of simulated single cells (n=1,600 for each sequencing depth) on a log-log scale. In the simulation setting, gene detection rate is higher with higher sequencing depth. (C) UMAP visualization for 1,600 simulated single cells derived from 16 tissues (100 cells for each tissue) with a sequencing depth of 1,000.

the PCA coordinates of the matched and normalized bulk RNA-seq samples from which the simulated single cells were derived from. Using these PCA-predicted coordinates we further predict the UMAP coordinates of the pre-trained UMAP model and plotted the results in Figure 4.10(left). Notably, the predicted location of the matched bulk samples locate in close proximity to the simulated single cells but not directly into the tissue clusters. In some cases, as for heart or stomach samples, the matched bulk and simulated single cells are closer than in other cases, as for nerve and lung samples. Assuming different technical characteristics between bulk and single cell samples cause the partial separation of tissue clusters, we initiated the next investigation. We now downsample the bulk references using library sizes of ~10,000 reads and used the pre-trained model to predict the coordinates in the PCA space as well as in the UMAP embedding following the procedure above (Fig. 4.10(right)). Now, the predicted locations of the downsampled (reference)

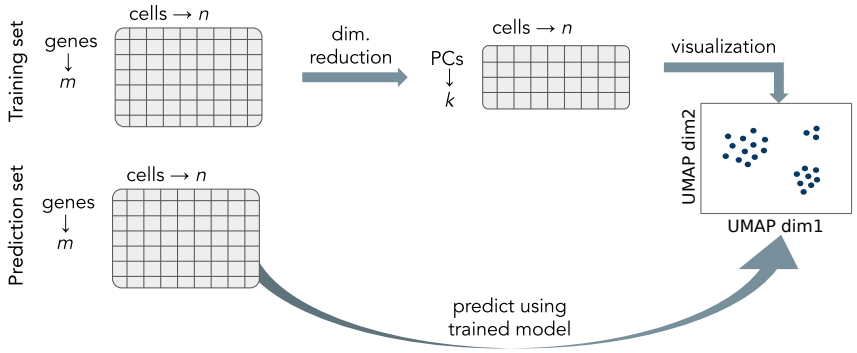


Figure 4.9: Concept of integrating data in a pre-computed embedding.
During training a set of single cells is used to reduce dimensions using PCA. Top-k principal components are used to compute UMAP coordinates for visualization. The prediction set consists of either bulk RNA-seq samples or other single cell samples excluded from training. Coordinates in the PCA space are predicted using the trained PCA model. Using the predicted PCA coordinates we further predict UMAP coordinates of the trained UMAP model.

single cells precisely fall into the corresponding tissue clusters.

Hence, bulk RNA-seq samples can be projected into the same embedding of a single-cell map. However, they do not necessarily appear directly in the cell cluster that they are associated with. As the simulated and matched bulk samples are biologically identical, the differences are solely due to the technical characteristics that cause the separation between the bulk and single-cell clusters from the same tissue. Using a downsampling approach on the bulk samples, it is possible to co-locate the bulk RNA-seq samples with the single-cell clusters.

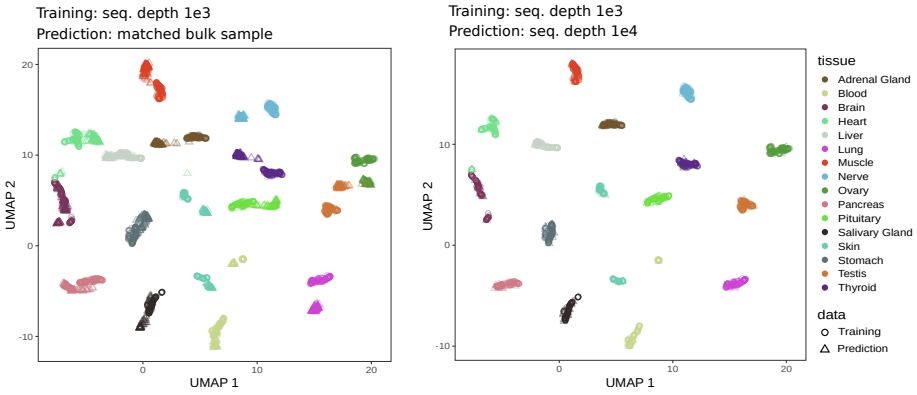


Figure 4.10: Integrating single-cell and bulk RNA-seq samples
For visualization, the dimensional reduction by PCA was trained on simulated single-cells with a library size of ~1,000 reads. The first 22 principal components were used to further train the UMAP embedding. The trained model was used to predict the location of bulk as well as downsampled single cells. On the right, we used the matched bulk samples of the simulated single cell to predict the location within the trained UMAP embedding. On the left, we generate new downsampled data (10 for each tissue) with a library size of ~10,000 reads. After downsampling bulk samples, the training and predicted data co-locate precisely in the UMAP embedding.

4.2.4 Assessing the technical variability in simulated single-cell count data

The above analysis initiates further investigation of the statistical features of the simulated single-cell data. Therefore, we randomly select a bulk reference RNA-seq sample and downsample 1,000 single-cells from this sample using a sequencing depth of 1,000 reads (Fig. 4.11). We reduce the dimensions by PCA and apply UMAP by calculating the embedding using the reference sample and the derived simulated single cells simultaneously. The projection shows a rounded-shape cloud formed by the simulated single cells along with the reference sample co-locating peripherally in the cloud. Note that the differences between the simulated samples are purely technical as they all derive from the exact same bulk sample. This provides a fundamental basis to investigate the cell-to-cell variation with respect to the technical variability discarding the influence of the biological variability.

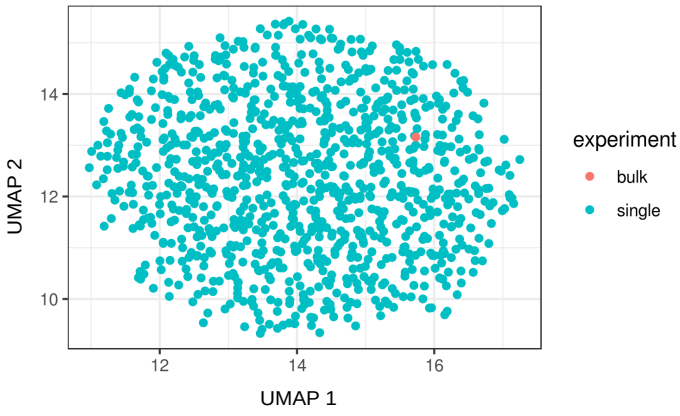


Figure 4.11: UMAP projection of tibial nerve derived single cells. 1,000 cells were derived and downsampled from the exact same bulk RNA-seq reference sample using a library size of 1,000 reads. UMAP embedding was calculated using the single cells and the reference sample together. The reference sample was randomly selected (`GTEX-SN8G-0426-SM-32PLF_Nerve_Nerve - Tibial`).

As Figure 4.11 shows, none of the samples are identical showing variations in the gene counts between the downsampled cells. Inspecting the data with regard to the technical variability we look at (i) the relation between the mean gene expression and variance across the cells and (ii) the relation between the dropout probability and mean gene expression in order to test the distributional hypotheses formulated in Section 3.1.4.

First, we plot the variance to mean relation for each gene across the simulated cells and plotted them in Figure 4.12(top). Observing a linear mean-to-variance relation with $\sigma^2 = \mu$ we assume a Poisson distribution and plotted a line with $y = x$ into the plot. Accurately the line stating a constant mean-to-variance relation characteristic for the Poisson distribution agrees with the observed data. As a next step, we plot the observed proportion of zero counts per gene across all cells in dependence of the mean gene expression in Figure 4.12(bottom). Remarkably the number of zeros observed in our data is highly dependent on the mean expression value and follows a sigmoidal curve where the amount of zero counts drops with a higher mean gene expression. Using Formula 3.3 we calculate the probability of observing zero counts under the assumption of an underlying Poisson distribution. We plot the calculated probabilities as a blue line. Precisely, the calculated probabilities and the observed zero counts match with one another, aligning with the assumptions about an underlying Poisson distribution. Note the sigmoidal curve

of the exponential function is due to the log-scaling of the x-axis.

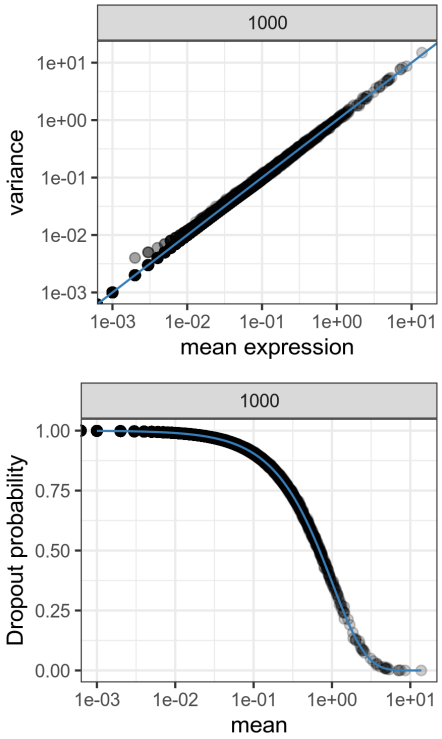


Figure 4.12: Technical variability follows a Poisson distribution.
1,000 cells were derived and downsampled from the exact same bulk RNA-seq reference sample using a library size of 1,000 reads. Above, mean expression and variance (log-log scale) for each gene is compared across all cells. The blue line ($y = x$) indicates the constant mean-variance relation in the Poisson distribution (Formula 3.2). Below, mean expression (log-scale) and dropout probability as the observed number of zeros is shown. Clearly, lowly expressed genes have a higher dropout probability. The mean expression refers to the intensity parameter λ of the Poisson distribution. The blue line represents the predicted dropout probability using $e^{-\lambda}$ (Formula 3.3).

We have shown that the variation due to downsampling from a multinomial distribution (the reference sample) follows a Poisson distribution as stated in Section 3.1.4. Here, we purely measure the technical variability across the samples as they have been derived from the same sample. However, in experimental RNA-seq datasets it has been observed that the variance increases more rapidly than the mean expression (Anders et al., 2010). Examples with the quadratic variance-mean relation of real single-cell RNA-seq data has been extracted from this blogpost by Svensson (`https://www.nxn.se/valent/2017/11/16/droplet-scrna-seq-is-not-zero-inflated`, downloaded 03/03/2022) and are depicted in Fig. 4.13. In these cases the assumptions of a constant mean-variance relation in a Poisson distribution are violated. Instead, a negative binomial distribution can be assumed allowing us to include and estimate the dispersion parameter to fit the data appropriately as pointed out in Section 3.1.4.

Considering the simulated dataset as well as the experimental single-cell RNA-seq count data we do not see any need to add a probability of observing zero counts in each gene by assuming a zero-inflated negative binomial distribution. The observed numbers of zeros in our simulated data can be modeled appropriately by the Poisson distribution solely. However, it does not account for biological variability as also previous studies have shown (Townes et al., 2019). In this case the negative binomial distribution is a better fit to model experimental single-cell RNA-seq data.

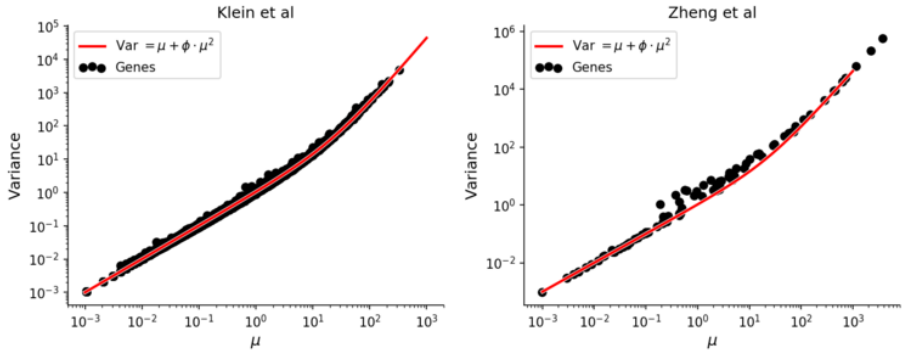


Figure 4.13: Quadratic mean-to-variance relation in experimental single-cell RNA-seq data.
Figure extracted from *Droplet scRNA-seq is not zero inflated —What do you mean "heterogeneity"?* (`https://www.nxn.se/valent/2017/11/16/droplet-scrna-seq-is-not-zero-inflated`, downloaded 03/03/2022) showing the experimental data following a negative binomial distribution. Red line represents the fit assuming a negative binomial distribution with an estimated dispersion parameter.

4.2.5 Studying cell similarities

The established simulation framework allows us to study technical differences between the simulated cells and their derived bulk references. In this section we want to investigate the similarities between the two the reference and its derived sample. Similarity measures provide an assessment to quantify how *similar* in terms of likeness or resemblance two objects are (Mulekar et al., 2017). Especially in single-cell RNA-seq data profiling many cells simultaneously, it is important to estimate which cells are more similar to each other than others in order to identify commonalities and hence cell identities (see Section 3.2.1). Equivalently, we can also talk about distance as the reciprocal metric for a similarity measure. Besides similarity and distance measures there exists *measures of association* as well. Measures of association for example Pearson's or Spearman's correlation coefficients are, strictly speaking, different from similarity measures in a formal perspective. They do not need to fulfill the criteria for a metric (defined below). However, for the sake of consistency and simplicity we use the term *similarity measure* in order to include all forms of "similarities" as an intuitive way.

There exists a large amount of measures in order to quantify similarities between two objects. In the following section we introduce some basics about different measures of similarity. We first start with the similarity measures representing a formal ***metric*** with d as a function on a set X with $d : X \times X \mapsto \mathbb{R}$ fulfilling the the following axioms $\forall x, y, z \in X$:

(i) $d(x, y) \geq 0$

(ii) $d(x, y) = 0 \iff x = y$

(iii) $d(x, y) = d(y, x)$ (symmetry)

(iv) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

We briefly summarize a few similarity measures that are used in this work:

Euclidean distance. The euclidean distance fulfills the metric criteria and is a popular measure used for continuous data. It is defined as the length of the line segment between two points x and y connecting them. In an n-dimensional space the following holds:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2} \tag{4.1}$$

Correlation coefficients. In statistics, there are two types to determine correlation coefficients: Pearson's correlation coefficient (Pearson's r) and Spearman's rank correlation coefficient (Spearman's ρ). Let X and Y be two random variables. Pearson's correlation coefficient is calculated as:

$$r_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{4.2}$$

with $\text{cov}(X, Y)$ as the covariance between X and Y, and σ_X and σ_Y the standard deviations of X and Y, respectively.
In contrast to Pearson's r, Spearman's rank correlation coefficient is calculated between the ranks of X and Y with rk_X and rk_Y, respectively:

$$\rho_{X,Y} = \frac{\text{cov}(\text{rk}_X, \text{rk}_Y)}{\sigma_{\text{rk}_X} \sigma_{\text{rk}_Y}} \tag{4.3}$$

Both types of correlation coefficients are not considered as a metric and therefore considered as measures of association. They can range from -1 to 1, interpreted as perfect anti-correlations to perfect correlation. However, Pearson's correlation coefficient assumes normally distributed data assessing linear associations whereas Spearman's rank correlation coefficient is non-parametric without any underlying assumptions on the data distribution. Hence, Spearman's rank correlation coefficient can be applied to any ordered data, continuous or discrete ordinal. In addition to that, Spearman's ρ is less sensitive to outliers as it considers solely the ranks of the variables in contrast to Pearson's r considering the actual values.

Shannon entropy and Kullback-Leibler divergence. In information theory the Shannon entropy is a measure of uncertainty and is defined as

$$H = -\sum_{x \in X} P(x) log(P(x)) \tag{4.4}$$

The Kullback-Leibler (KL) divergence, also known as the relative entropy, between two probability distributions P and Q is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log\left(\frac{P(x)}{Q(x)}\right) \tag{4.5}$$

The Shannon entropy is a special case of the KL divergence with the probability distribution of the sample P and the probability distribution Q as the uniform distribution. The Kullback-Leibler divergence is asymmetric because generally $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. However, a symmetric measure can be obtained by the sum of the two asymmetric measures: $D_{KL}^{sym} = D_{KL}(P||Q) + D_{KL}(Q||P)$.

Similarities between the reference and its derived single cell sample

Given the above introduced similarity measures, we now investigate the similarities in our simulated data. We model the simulated single cells in a downsampling procedure derived from their bulk references that presume a multinomial distribution. Hence, we can consider two probability distributions and use the Kullback-Leibler (KL) divergence with P as being the single-cell sample and Q its corresponding reference sample. In our simulation scenario we derive single-cell count data from a bulk reference sample using a sequencing depth of ~1,000 and ~10,000 reads. We can now use the reference sample and each of its derived single-cell samples to compare the similarities with one another using the KL divergence (Fig. 4.14).
First, we look at the general KL divergences in the simulated dataset with a sequencing depth of 1,000 and 10,000, respectively (Fig. 4.14A). Interestingly, we get different value ranges with respect to the KL divergences between the low and high sequencing depth simulation scenarios. The KL divergences in the low sequencing depth dataset have a considerably higher KL divergence values with a stronger variation than in the high sequencing depth dataset. Next, we divide the KL divergences into their corresponding originating tissue — distinguishing between low and high sequencing depth data (Fig. 4.14B). We can clearly see that different tissues have different KL divergence distributions that are very similar across the two simulated sequencing depth scenarios. However, as Figure 4.14A indicates, the corresponding mean values of the KL distances in the respective tissue distributions are larger for single cells with a low sequencing depth.

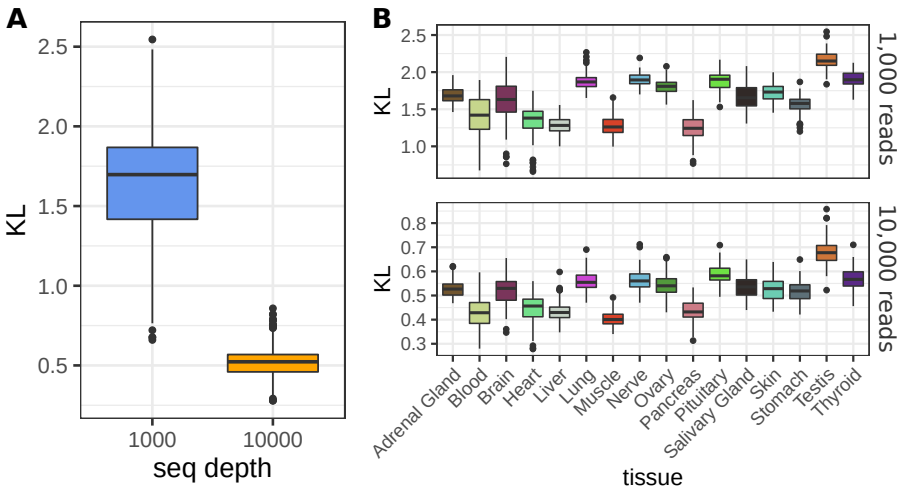


Figure 4.14: Kullback-Leibler divergence between reference and single-cell derived sample.
(A) Lowly sequenced simulated cells have a generally higher Kullback-Leibler (KL) divergences than simulated cells with a larger sequencing depth. (B) KL divergences across tissues for simulated cells with a ~1,000 read library size (top) and a ~10,000 read library size (bottom). Note, that the value ranges between the y-axis are different.

We additionally derive Pearson's correlation coefficient between the reference and its corresponding single cell. As a measure of association Pearson's correlation coefficient estimates the linear relationship between the reference and its derived single cell sample. Similar to the KL divergence

we see distinct distributions for the two simulation scenarios with higher Pearson's correlation coefficients in larger sequencing depths (Fig. 4.15A). Here, cells with a sequencing depth of ~10,000 reads lead to almost perfect Pearson's correlation with only little variation. However, similarity scores obtained from cells with a lower sequencing depth range from roughly 0.6 to almost perfect correlation. Examining the similarity score distributions within the tissues we see large differences for both simulation scenarios (Fig. 4.15B). Tissues such as heart, liver and pancreas have very high correlation coefficients whereas lung, nerve and ovary have considerably lower values in the respective sequencing depths scenarios. However, similarly to the tissue distributions of the KL divergences the correlation distributions do not differ much for the respective tissues albeit from the different sequencing depths simulation.

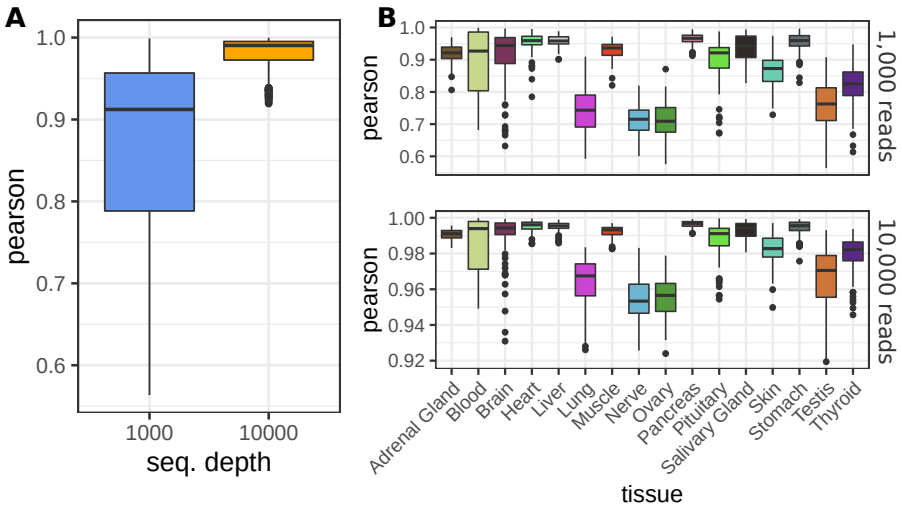


Figure 4.15: Pearson's correlation coefficient between reference and single-cell derived sample.
(A) Lowly sequenced simulated cells have a generally lower correlation than simulated cells with a larger sequencing depth. (B) Pearson's *r* across tissues for simulated cells with a ~1,000 read library size (top) and ~10,000 read library size (bottom). Note, that the value ranges between the y-axis are different.

In conclusion, we observe that similarity measures are strongly dependent on the sequencing depth. Here, larger sequencing depths lead to more similar values with respect to its corresponding reference sample. However, we see a remarkable difference in the distribution of similarity measures across tissues that are reproducible with different sequencing depths. Even the very low number of ~1,000 reads per cell preserve the distribution of Pearson's correlation values within each tissue. Hence, we observe a robust tissue-specific signal across varying sequencing depths.

Cell similarity and gene detection rate concordance

The two similarity measures, KL divergence and Pearson's correlation coefficient have a reciprocal relation: Samples with high KL divergences tend to generally have a low Pearson correlation (Fig. 4.16A). Considering the tissues, there are differences in how the samples spread across the similarity measure ranges: For example, muscle, pancreas and liver samples span only a short

range of Pearson correlation coefficients with a relatively large range of KL divergences values. As opposed to that, testis, nerve, lung or thyroid derived samples have a relatively small span in KL divergences but a large range of Pearson correlation coefficients. Hence, the above obtained similarity measures between the reference and its derived single cell emphasize a tissue or sample specific dependency.

Further, we look at the gene detection rate per sample and compare it to the corresponding Pearson's correlation coefficient (Fig. 4.16B). Interestingly, we see a similar trend between the two measures as compared to the relation between the KL divergence and Pearson's correlation coefficients: Generally, higher gene detection rates go along with a low Pearson correlation. However, we do see again some tissue-specific dependency influencing the variation of the Pearson correlation. We can explain the same trend between the KL divergence and detection rate, respectively by the Shannon entropy assessing the level of uncertainty. Hence, the Shannon entropy of a sample is higher for higher gene detection rates (Fig. 4.16C). For this reason we see a similar tendency between the KL divergence and the detection rate to the respective Pearson correlation. However, it still remains of a particular concern that a similarity measure is coupled with a sample's detection rate.

The above analyses use the reference bulk sample as a ground truth in order to evaluate similarity measures from its derived single cell. Our findings show that given a sequencing depth there is a tissue/sample-dependent factor affecting similarity scores between the reference and its derived sample. Furthermore, we see that the presented similarity measures are highly dependent on the detection rate. Subject to these considerations one needs to account for these technical factors in order to appropriately estimate similarities.

4.3 DISCUSSION

This chapter gave us a first glimpse of single-cell RNA-seq data along with its challenges in particular with regard to its noise level and the cell-to-cell variability.

The first Section covered an experimental dataset with Influenza A virus infected cells characterized by cells with a high and low viral replication rate. We made an attempt to explain the heterogeneity within these cells. However, the dataset was marked by a high technical noise level measured by external spike-ins masking the biological signal. In addition, high expression levels from the viruses make it difficult to read out the cellular gene expression. For this reason, it was not possible to find clear evidences from the cells' gene expression explaining the cell-to-cell variability. Nonetheless, our results have shown an association between the viral replication rate and defective interfering particles originated from the viruses. These results are presented in the research study originated by Kupke et al., 2020.

The second Section covered a simulated single-cell dataset generated from the GTEx database — a public bulk RNA-seq data resource for human tissues. We implemented a simulation framework allowing us to generate reference-derived single-cells. Using the GTEx derived single cells we

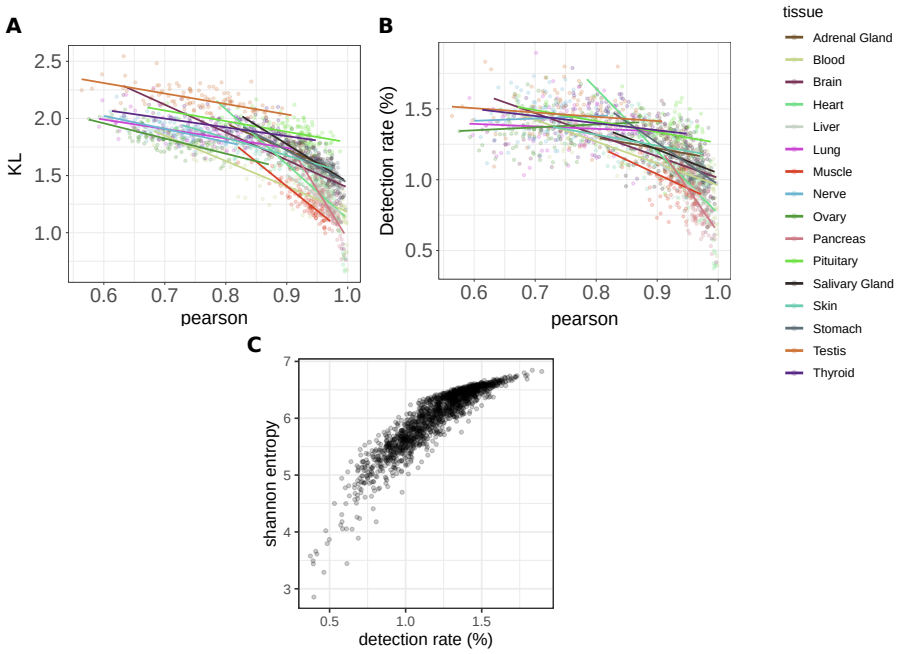


Figure 4.16: Gene detection rate associated with cell similarities.
(A) Relation between Kullback-Leibler (KL) divergence and Pearson's correlation coefficient in single cells with ~1,000 reads. Measures were obtained between the reference and its derived simulated single cell, respectively. (B) Gene detection rate and Pearson's correlation coefficient. Gene detection rate was measured in the simulated single cells (CPM > 0). Lines represent linear regressions of the corresponding tissue-derived samples (color-coded) for both plots. (C) Shannon entropy and gene detection rate scatterplot show good correlation with one another.

could do multiple investigations *in silico*: We first show how single-cell and bulk RNA-seq samples can be integrated into the same embedding. By training a pre-computed embedding bulk RNA-seq samples can be integrated into the topological map. Here, similar sequencing depths (e.g. by downsampling bulk samples) to the samples used for training yield more accurate co-locations within the embedding. In the future this framework can be used to co-locate single-cell experiments with bulk RNA-seq experiments. This allows us to characterize and annotate cell clusters without searching for marker gene expression. Certainly, if the two experimental dataset used for training and prediction arise from different sources (different labs, time points etc.) an appropriate preprocessing procedure including normalization and batch correction needs to be performed prior to the data integration. Using the simulated framework we further derive multiple single cells from the same bulk RNA-seq sample and examined the variability across the simulated cells. As the cells were drawn from the same reference sample they only differ by their technical variability enabling us to assess the variability across the cells. We are able to fit the technical variability of the simulated cells by a Poisson distribution characterized by a constant mean-to-variance relation. Lastly, we evaluated similarity measures between a reference bulk and its derived single-cell sample and observed a high dependence between the gene detection rate and similarity measures such as Pearson's correlation coefficient.

This chapter explored technical factors within single-cell transcriptome data and had a focus on measuring the technical variability and differences between the bulk RNA-seq samples and the derived single cells. The next chapter aims the attention at the observed biological heterogeneity in single-cell transcriptome data. It examines the data with an analytical viewpoint inferring associations between genes with regard to their regulatory interactions using network reconstruction.

5 FROM TRANSCRIPTOME DATA TO GENE REGULATORY NETWORKS

Biological experiments measuring transcription profiles represent a snapshot of the underlying biological process. By selecting samples from different time points one might get a picture of how samples, i.e. cells, evolve. In fact, the identity of each cell is determined by a biological process that is carefully regulated by certain factors. These factors, called transcription factors, regulate and control cell differentiation and drive transitions from one cellular state to another or maintain cell identities. A powerful tool in systems biology is the use of gene regulatory networks (GRNs) in order to study gene interactions that drive biological processes. As GRNs are often unknown, researchers *reconstruct* or *infer* the underlying network from high-throughput data such as RNA-seq data. In this process the cellular heterogeneity plays an important role. In order to study gene regulatory networks a certain level of cell-to-cell variability is needed to explain and understand the heterogeneity across the cells during dynamic processes. Otherwise, with cells being very homogeneous and only differing by their technical variance, the content for biological information is very low.

This chapter investigates the use of variability in single cell transcriptome data to reconstruct GRNs. I first give an introduction about the mathematical fundamentals in network theory and further present state-of-the-art algorithms to reconstruct GRNs. Then, I introduce an algorithm called *neighborhood selection* (Meinshausen et al., 2006). I use a methodology based on that concept and adapted the algorithm to infer GRNs using transcriptome data. To test the implementation, I applied our method to *in silico* data that was generated with a known network structure. The networks has been designed in such a way that they simulate common differentiation scenarios observed in systems biology from simple linear differentiation paths to complex differentiation paths branching into different cell states. Furthermore the data allows us to dynamically reconstruct networks along the differentiation path and investigate how the reconstructed networks change upon time-ordered transcriptome data. Additionally, I apply our method on experimental single-cell RNA-seq data sampled from hematopoietic stem cell differentiation in mice to test the prediction performance on real data.

5.1 MATHEMATICAL PREREQUISITES

A gene regulatory network can be represented as a graph G. In graph theory, a graph is defined as a pair (V, E) with V as a set of vertices or nodes, and E as a set of edges. In GRNs nodes represent genes and an edge their interaction with one another.
Edges can be either undirected or directed. Directed edges have a source and a sink node, where the source node usually represents the regulator, also known as a ***transcription factor***, and the sink node represents the ***target gene***. Furthermore, there are two types of interactions between genes. They can be either activating or inhibiting interactions. While the former activates the

target gene which is indicated by an increase in the target gene's expression level, the latter inhibits the target gene's activity which is indicated by a decrease in the target gene's expression level.

There are several terms characterizing a network that can be formulated as follows: the ***size*** of a network simply corresponds to the number of vertices within a network.
The ***network density*** includes the number of edges and captures the number of edges in relation to the total number of possible edges. In an undirected network this corresponds to:

$$d = \frac{|E|}{|V| * (|V| - 1)/2} \tag{5.1}$$

Here, the denominator is to the binomial coefficient $\binom{|V|}{2}$. Furthermore, the ***node degree*** corresponds to the number of outgoing edges from a source node. Nodes with a high node degree are particularly interesting as they usually represent highly connected nodes, so-called hubs. Those hubs serve as potential key factors regulating important biological processes.

5.1.1 Network data structures

In graph theory and as computer–readable format there are two common data structures to represent a network:

Adjacency matrix. Given a graph $G = (V, E)$ the adjacency matrix A is a $|V| \times |V| = n \times n$ dimensional matrix :

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

with entries $a_{ij} = 1$ if there is an edge between node i to node j, and $a_{ij} = 0$ if there is no edge. In case the graph has weights $a_{ij} \neq 0$ for each edge e ∈ E. The adjacency matrix is symmetric if the graph is undirected with $a_{ij} = a_{ji}$.

Adjacency list. As an alternative, a graph $G = (V, E)$ can be represented by an adjacency list L which is an array of length $|V| = n$ comprising $|E| = m$ elements. The i-th entry of L is the set comprising the neighbors of node i. In case the graph has weights, the set consists of tuples (v_j, w), with w being the weight of the edge between node i and node j.

5.1.2 Problem definition

Given a gene expression dataset M represented by a $N \times P$ matrix with N samples (here cells) and P genes. A row vector n with $n = 1, ..., N$ represents an N-dimensional vector with gene expression values for each cell, and a column vector y with $y = 1, ..., P$ represents a P-dimensional vector with gene expression values for each gene in the respective cells. The research aim is to infer the underlying network from the given gene expression matrix M. The predicted GRN

consists of a set of regulatory interactions between any two genes from the total of P genes assembling to a graph with P nodes. The regulatory interactions can either represent a direct physical association or an indirect association between two genes. Indirect regulation may arise from transitive associations where gene A is associated with gene B, and gene B is associated with gene C. As a consequence, the indirect association between gene A and gene C oftentimes arises undesirably. Ideally, GRNs are free of indirect regulations. However, this remains a challenging task as we will see in the matter of this chapter. In the following we summarize a few approaches to reconstruct GRNs from transcriptome data.

5.2 STATE-OF-THE-ART ALGORITHMS TO RECONSTRUCT GENE REGULATORY NETWORKS

Algorithms to reconstruct GRNs go back to mid-2000s and were first developed for population based samples such as micro arrays or later bulk RNA-seq experiments. There are two extensive benchmark studies summarizing and evaluating the performance of individual GRN reconstruction algorithms (Chen et al., 2018b; Pratapa et al., 2020). In this section we introduce a few algorithms with their rough mathematical concepts divided into information, correlation, regression-based approaches. Based on the their concept, the methods provide an (un-)directed network, either with or without the information about the mode of the potential interaction. Table 5.1 provides a short overview of a few selected algorithms. Note that the selected algorithms are not an exhaustive list of published tools but rather give a broad overview about possible concepts for network reconstruction in gene expression data, both for bulk RNA-seq experiments as well as specifically designed for single-cell RNA-seq data. We selected the algorithms based on good performances that was evaluated by a benchmark study published by Pratapa et al., 2020.

Table 5.1: Tools for reconstructing gene regulatory networks. Selected algorithms ordered by the underlying mathematical approach and year. Column 'Time' indicates whether temporal ordered cells are required. Columns 'Directed' and 'Interaction type' refer to the output network whether the edges have a direction and the information if there is an activation or inhibition.

Tool	Author, Year	Approach	Time	Directed	Interaction type
PIDC	Chan et al., 2017	information theory	no	no	no
PPCOR	Kim, 2015	correlation	no	no	yes
GENIE3	Huynh-Thu et al., 2010	regression	no	yes	no
GRNBoost2	Moerman et al., 2019	regression	no	yes	no
SINCERITIES	Papili Gao et al., 2018	regression	yes	yes	yes

5.2.1 Information theoretic approaches

Oftentimes, information based approaches use the ***mutual information (MI)*** as a measure of association. The MI is a pairwise measure which determines the degree of statistical dependency between two random variables (i.e. genes) X and Y. It is defined by

$$\begin{aligned} MI(X, Y) &= \sum_{i,j} P(x_i, y_j) log \left(\frac{P(x_i, y_j)}{P(x_i)P(y_j)} \right) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned} \tag{5.2}$$

where $P(x_i)$ and $P(y_j)$ are the marginal probabilities of x_i and x_j, respectively, $P(x_i, y_j)$ is the joint probability distribution over X and Y, and H(X) is the Shannon entropy (see Eq. 4.4). Note that the MI is linked to the previously introduced Kullback-Leibler divergence as follows.

$$MI(X, Y) = D_{KL}(P_{(X,Y)} || P_{(X)} \times P_{(Y)}) \tag{5.3}$$

Here, the product of the marginal probabilities are compared to the joint probabilities. Thus, MI values of zero values represent an equal probability between the joint and the product of the marginal probabilities. Conclusively, X and Y are independent from each other which means the observation of X does not inform anything about Y.

PIDC (Chan et al., 2017). This algorithm is based on partial information decomposition (PID) and explores for every three genes g_1, g_2 and g_3 their statistical dependencies to one another. Here, the mutual information between g_1 and g_2, given the third gene g_3 is estimated by partitioning the mutual information into redundant, synergistic and unique contributions. More specifically, the algorithms uses the ratio between the unique contribution and the mutual information for each gene (i.e. g_1 and g_2) conditioned on the third gene g_3. By iterating the third gene across all remaining genes in the network and subsequently taking the sum of these ratios the *proportional unique contribution* is estimated. By this, every triplet gene combination is considered. Finally, PIDC detects the most important interactions per gene using a threshold that depends on the underlying distribution and confidence score. As the proportional unique contribution is symmetric PIDC provides an undirected graph.

5.2.2 Correlation based approaches

We determined earlier the measure of pairwise correlation values (Section 4.2.5) that a few algorithms make use of. Generally, correlation values range from -1 to 1. The signs of the resulting correlation values can be interpreted as inhibitory and activating regulations in gene regulatory networks.

PPCOR (Kim, 2015). As a correlation based approach PPCOR employs the concept of partial or semi-partial correlation in order to infer the network structure. Partial or semi-partial correlations are a mean of association between two genes considering the effect of all or a subset of the remaining genes, respectively. The package provides p-values and further statistics in order to

estimate the level of significance between the potential associations. As a symmetric measure and by using the signs of the correlation values, PPCOR provides an undirected graph with inhibitory and activating associations.

5.2.3 Regression based approaches

Regression based approaches use the expression pattern of a set of genes (features) in order to predict the expression pattern of a target gene (response). The underlying prediction model matters if the resulting network includes signed or directed interactions.

GENIE3 (Huynh-Thu et al., 2010). Initially developed for bulk transcriptome data, GENIE3 uses a tree-based ensemble method with Random Forest in order to predict the gene expression pattern of the target gene using all remaining genes as predictors. Iteratively, every gene is used as a target gene in each random forest model. The importance of the predictors are collected and aggregated over all runs. This provides a ranking of possible interactions from which the directed gene regulatory network is reconstructed from.

GRNBoost2 (Moerman et al., 2019). This approach uses the principle of GENIE3 and is a fast alternative on datasets with large sample sizes, e.g. in single-cell RNA-seq data. It uses gradient boosting machine with regularized stochastic variation in order to increase computational efficiency. As a derivation of GENIE3 it provides a gene regulatory network with directed interactions ranked by their importance.

SINCERITIES (Papili Gao et al., 2018). Using time-stamped gene expression data, this method uses regularized linear regression models (ridge regression) in order to recover directed interactions from gene expression data. Here, the changes in expression of one gene in a given time interval are used to predict the changes in the gene expression of another (target) gene in the next time interval. Therefore, Granger causality concept is applied providing directed regulations among genes. Furthermore, it uses partial correlation values in order to determine the mode of interaction (activating or inhibiting).

5.3 NEIGHBORHOOD SELECTION TO RECONSTRUCT GENE REGULATORY NETWORKS

Gene regulation drives biological processes such as cell differentiation and the maintenance of cell identity. Thus, revealing the underlying gene regulatory landscape using graph structures in single-cell transcriptome data is an active field of research. We introduced a few algorithms attempting to infer graph structures. Mainly, they differ in their main mathematical concept but also in prerequisites such as (pseudo) time information of the data. Additionally, the estimated network differs with regard to the interaction type and causality of gene regulation. Benchmark studies published by Chan et al., 2017 and Pratapa et al., 2020 provide extensive evaluation results

of the current available methods.

In addition to the available methods we reconstruct gene regulatory networks using a regression-based approach called ***neighborhood selection***. This algorithm infers graph structures in high-dimensional data using lasso regression (Tibshirani, 1996) and was developed by Meinshausen and Bühlmann in 2006 (Meinshausen et al., 2006). In neighborhood selection the algorithm decomposes the prediction of adjacent genes (neighborhood) of P target genes into P separate regression problems. Here, the lasso regression enables feature selection via a shrinkage parameter λ that specifies the size of the neighborhood. Hence, adjacent genes in a neighborhood are conditionally dependent given all remaining genes. In turn, non-adjacent genes (without an edge) are conditionally independent given all remaining genes in the network.

5.3.1 Mathematical background

Before we show how we use neighborhood selection in order to reconstruct gene regulatory networks we introduce some linear algebra underlying the mathematical basics. Here, we mainly extract the basics from the textbook "The Elements of Statistical Learning" (Hastie et al., 2009) and Prof. Martin Vingron's lecture notes "Construction of Biological Networks" from Summer 2019.

Gaussian graphical models. Let us assume our data comes from a multivariate Gaussian distribution with random variables $X_1, ..., X_P$ and mean μ and a covariance matrix Σ:

$$X = (X_1, ..., X_P) \sim \mathcal{N}(\mu, \Sigma) \tag{5.4}$$

The graphical model is a graph $G = (V, E)$ that explains the statistical dependencies among the variables in the data matrix. Here, statistical independence in the context of probability theory means:

$$P(A|B) = P(A) \text{ and } P(B|A) = P(B) \tag{5.5}$$

with $P(A|B) = P(A, B)/P(B)$. Hence, conditioning on either A or B has no influence on the respective probability. In context of the graphical model two variables X_i, and X_j are conditionally independent if the respective entry of the covariance matrix Σ is 0.

Partial correlation. In Gaussian graphical models we define *partial covariances* as the degree of association between X_i and X_j, conditioned on all remaining random variables in the data. *Partial correlation*, in turn, is the scaled version of partial covariance. It can be derived by studying a prediction problem where we aim to estimate any response vector Y from the data X:

$$Y \sim \beta^T X \tag{5.6}$$

with β being the coefficients to obtain estimates $\hat{Y}$. We now want to calculate the partial correlation between X_i and X_j by fitting two linear models:

1. $X_i \sim \beta^T X \setminus \{X_i, X_j\}$

2. $X_j \sim \beta^T X \setminus \{X_i, X_j\}$

Now, the partial correlation refers to the Pearson correlation of the residuals arising from the two linear regression models:

$$\rho_{X_i,X_j} = cor(X_i - \hat{X}_i, X_j - \hat{X}_j) \tag{5.7}$$

Here, the partial correlation is 0 if and only if the two random variables are conditionally independent given all remaining variables.

Inverse covariance matrix. For Gaussian data the entries of inverse of the covariance matrix P = Σ^{-1} (also known as precision matrix or concentration matrix) refer to the partial correlation of any two pairs of random variables X_i and X_j in the set of nodes V:

$$\rho_{X_i,X_j|rest} = -\frac{p_{ij}}{\sqrt{p_{ii}p_{jj}}} \tag{5.8}$$

Hence, if the entry (i,j) in Σ^{-1} is 0 X_i and X_j are conditional independent given all remaining variables. Conclusively, for Gaussian data missing edges in the graphical model refer to a conditional independence between a pair of variables with respect to the remaining variables.
However, it is not trivial to invert Σ^{-1} especially if the data matrix is singular (not full rank) or ill-conditioned. There exist many approaches estimating Σ^{-1} to obtain the graph structure. We use a lasso regularization introduced by Meinshausen et al., 2006 that delivers an approximation of the graph explaining the underlying data.

5.3.2 Lasso regularization for graph estimation

Lasso (least absolute shrinkage and selection operator) regularization is commonly used in linear regression models (Tibshirani, 1996) to select a set of features explaining the response or target vector Y. The lasso estimate of a regression problem is defined as

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{P} |\beta_j| \right\} \tag{5.9}$$

with y_i being the entries of the response vector, x_{ij} being the entries of our data matrix, β being the coefficients to be estimated, λ the shrinkage parameter, and $\sum_i^P |\beta_j|$ being the L_1 lasso penalty. Similarly to a regression problem, we estimate the coefficient of our linear models $\hat{\beta}$ with an intercept β_0 by minimizing the prediction error between the model and the response vector Y. However, in the lasso regularization, an additional parameter λ is added controlling the strength of the shrinkage of the regression coefficients β. The higher the value of λ, the higher the shrinkage which typically results in many coefficients in β being equal to 0. Thus, the number of non-zero coefficients and hence the number of included features is controlled by λ.

Meinshausen and Bühlmann adopted the idea of Lasso regularization in the context of network reconstruction (Meinshausen et al., 2006). The authors introduced neighborhood selection and defined a neighborhood ne_i of a node $i \in V$ as the smallest subset of $V \setminus \{i\}$ such conditioned on variables X_{ne_i} in the neighborhood ne_i, X_i is conditionally independent of all remaining variables.

The algorithm determines the neighborhood of node i by fitting a lasso regression using the variable X_i as the response vector and the remaining variables as features. Given any shrinkage parameter λ the lasso regression estimates the coefficients of β. Different shrinkage-levels of λ lead to a denser (low λ values) or to a sparser (high λ values) network structure (Fig. 5.1). The entries in β_j reflect the conditional (in)dependence between nodes i and j. However, an edge can be drawn either if the respective entries from node i to j *and* from node j to i are non-zero (logical AND ($\wedge$) operator), or if any of the entries is non-zero (logical OR ($\vee$) operator).

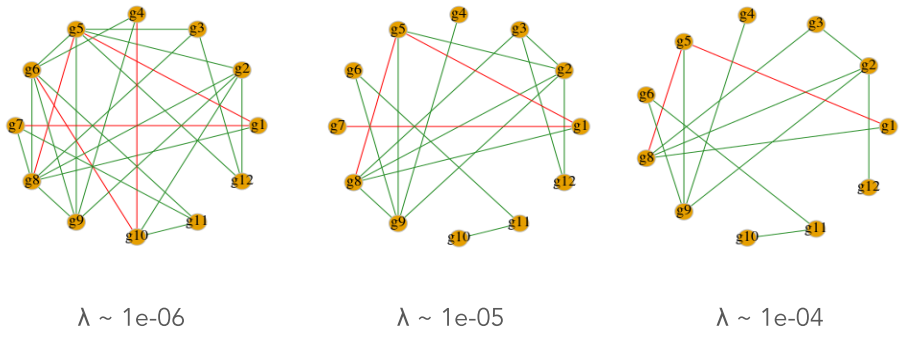


Figure 5.1: Varying λ influences the network structure.
Example network with 12 genes. We apply neighborhood selection with different λ parameters and plot the resulting network. Red edges are negative correlations and green edges positive correlations representing inhibiting or activating regulations. Increasing λ results in sparser networks.

5.4 GENERATING *IN SILICO* DATA

Similar to Section 4.2.1 we need to have ground truth data to verify the proposed method. Here, we need gene expression data that are derived from a known regulatory network. To this end, we generate *in silico* data using the tool BoolODE (Pratapa et al., 2020). BoolODE takes as input a Boolean network which is dependent on Boolean rules and models stochastic time-course data based on an ODE approach. This provides an optimal evaluation as the networks are known and have a decent amount of nodes and interactions that are feasible to control. BoolODE covers a set of synthetic networks as well as literature-curated networks. The synthetic networks model different scenarios occurring in single-cell biology such as cyclic, linear or diverging lineages. The literature-curated networks are derived from well-studied cell differentiation processes with known and validated interactions gathered from different studies.

In this chapter we analyze three different synthetic scenarios simulating linear, cyclic and multifurcating differentiation paths. Additionally, we examine a literature-curated network that simulates myeloid blood differentiation giving rise to erythrocytes, megakaryocytes, monocytes and granulocytes. The underlying Boolean networks have different degrees of complexity with regard to the number of genes and their regulations. This allows us to evaluate the reconstructed

networks obtained by the method on rather simple scenarios (linear) to very complex biological systems.

5.4.1 From network models to simulated data

A Boolean network is an easy way to describe gene regulatory networks. Here, the nodes (genes) have a binary state which is either 'ON' or 'OFF'. Depending on the mode of interaction, which is either activating or inhibiting, the state of the regulator's target gene can switch over time. Hence, it is a dynamic way to explore the network's status in a time-dependent manner. The small network depicted in Figure 5.2 provides us a simple example:

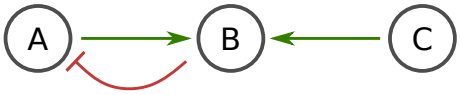


Figure 5.2: Example of a regulatory network. The regulatory network consists of three genes A, B and C. Green arrows depict activating, red arrow depicts inhibiting interactions.

We can now derive Boolean rules using the logic operators $\wedge$ (AND), $\vee$ (OR), and $\neg$ (NOT), representing the regulations among the genes in the above network:

$$\begin{aligned} B_{(t+1)} &= A \wedge C \\ A_{(t+1)} &= \neg B \end{aligned} \tag{5.10}$$

Based on the Boolean rules we define each status of the networks by using all possible 'ON'/'OFF' or 1/0 combinations in order to determine the outcome of the network. Oftentimes truth table are used for that purpose. Table 5.2 shows the truth table for gene B in Equation 5.10:

Table 5.2: Truth table for a Boolean rule w.r.t. gene B.

A	C	B
0	0	0
0	1	0
1	0	0
1	1	1

Once we define the set of rules and their respective truth tables, we start to model the dynamic system. The dynamic system needs to consider two aspects: The mRNA level and the protein level. As described earlier (see Section 2.1), the mRNA level of a gene depends on the transcription rate as well as its degradation rate. Upon mRNA transcript synthesis the mRNA needs to be translated into a protein in order to regulate the target genes. Hence, the translation rate and its

protein degradation rate need to be modeled accordingly. BoolODE models the mRNA level and the protein level using the following system of coupled differential equations:

$$\begin{aligned} \frac{d[x_i]}{dt} &= mf(R_i) - l_x[x_i] \\ \frac{d[p_i]}{dt} &= r[x_i] - l_p[p_i] \end{aligned} \tag{5.11}$$

with m as the transcription rate and l_x the degradation rate of gene x_i, r as the translation rate of gene x_i and l_p the degradation rate of protein p_i. $f(R_i)$ are the functions including the regulations defined in the Boolean rule. Following Pratapa et al., 2020 we use the Hill equation which is a sigmoidal function modeling the respective concentrations of the gene products. Hence, considering only the true value of gene B in our truth table 5.2 and assuming equal concentration parameters for each gene product, gene B is modeled by the differential equation:

$$\frac{d[B]}{dt} = m\left(\frac{[A][C]}{1 + [A] + [C] + [A][C]}\right) - l_x[B] \tag{5.12}$$

The procedure has previously been used to generate *in silico* data from Boolean network models (Schaffter et al., 2011). BoolODE's realization adds a stochasticity term to the differential equation for noise amplification according to gene expression data. Given a set of kinetic parameters describing the different rates of the ODE models (Tab. 5.3) and a set of initial conditions describing which gene is initially active the tool starts stimulating stochastic time-course data. Here, in the model simulation a vector of gene expression values corresponds to a particular time point within the simulation procedure which refers to a single cell. By this, the provided *in silico* data models the dynamic system in a time-course dependent manner.

Table 5.3: Kinetic parameters used in BoolODE.

Parameter	Symbol	Value
mRNA transcription rate	m	20
mRNA degradation rate	l_x	10
Protein translation rate	r	10
Protein degradation rate	l_p	1
Hill threshold	k	10
Hill coefficient	n	10

5.4.2 Simulation results

We use BoolODE to depict three different scenarios commonly observed in biological systems: linear, cyclic and multifurcating (here trifurcating) trajectories (see Fig. 5.3). The linear trajectory

is characterized by a single path where cells traverse directly from early time points to late time points. The underlying network model (Fig. 5.3, top row) is a cascade of genes activating one another using an initial state condition of $g1 = 1$ starting the cascade. The cascade of sequential gene activation can be noticed in the gene expression heatmap (Fig. 5.3, middle row). Genes get activated successively similar to a domino effect. Once g7 is activated it inhibits g1 which turns off the subsequent genes. The cyclic trajectory forms a single path where the initial state merges with the final state arranging a circle. Similarly to the linear network model, the cyclic model consists of a cascade of genes with two consecutive inhibiting interactions between g1 – g2 – g3 which is followed by two activating interactions between g3 – g4 – g5. Upon activation of g5 it inhibits g1 which in turn activates g6. The initial state conditions are set to $g1 = g2 = g2 = 1$ to start the simulation. The pairwise associations seen in the gene expression heatmap reflect the underlying interactions: While g1 is expressed it prevents g2 from being expressed which, in turn, keeps g3 in an active state. The successive activating interactions between g3, g4 and g5 have the same cascade expression pattern observed in the linear trajectory. The expression pattern of g6 can be traced back by its activation from g1. This model creates the cyclic trajectory visualized in Fig. 5.3, bottom middle panel. A common example for cyclic trajectories is the cell cycle process. Lastly, the trifurcating trajectory describes a process where cells start from a common initial cell state and branch into three different final cell states. While the first two underlying network models are simple to understand due to the small number of interactions, the underlying network modeling the trifurcating trajectory is more complex having genes with more than one interaction partner. For some of those genes, e.g. g5, have self-loops activating the genes itself as well as bidirectional types of regulations. Using an initial state condition with $g1 = g7 = 1$ the network models a trajectory with three distinct cell states. Multifurcating trajectories are common during cell differentiation processes with multiple cell fates.

Given the input Boolean network we apply BoolODE to generate *in silico* data. We use t-SNE to visualize the trajectories that are modeled by the underlying reference networks (Fig. 5.3). Clearly, the gene expression data follow the three expected trajectories that we aim to model. Early to late cell stages are depicted in the color code. Hence, BoolODE successfully generates time-dependent gene expression data that we can use to reconstruct the gene networks. By comparing the predicted network with the actual underlying network, we can evaluate the performance of the neighborhood selection method.

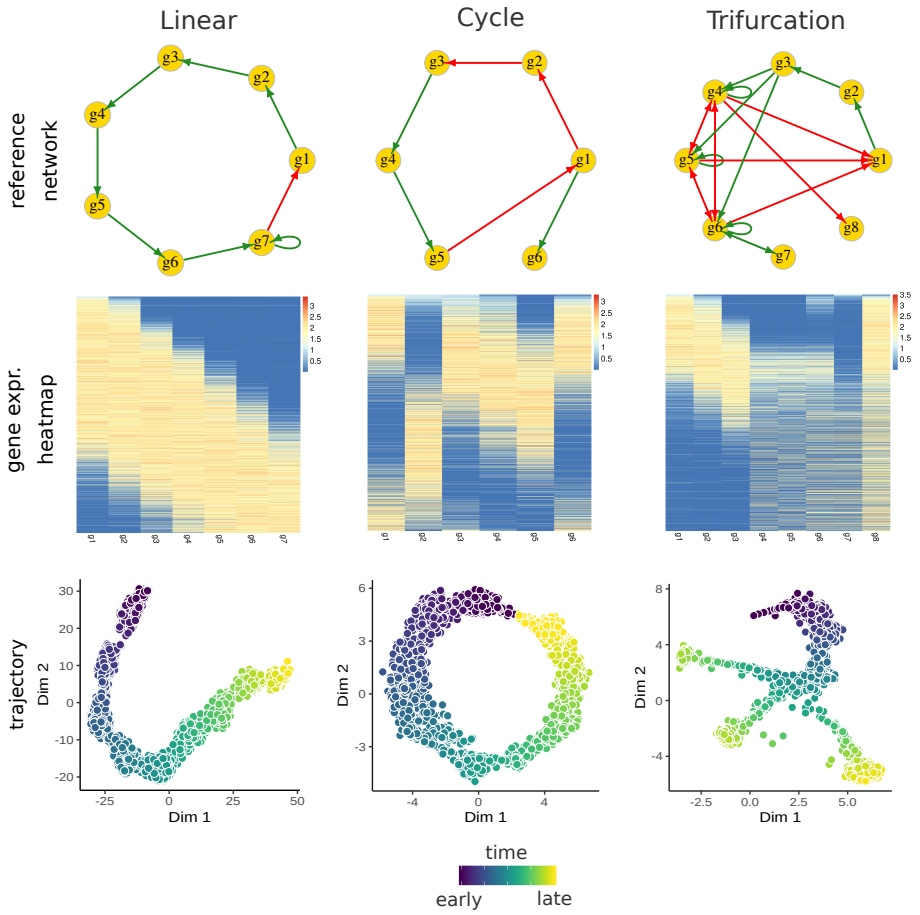


Figure 5.3: Boolean networks and their corresponding time-course *in silico* data.
Boolean networks modeling linear, cyclic and trifurcating lineage trajectories. Green arrows represent activating regulations, red arrows represent inhibiting regulations. The heatmap shows gene expression values for each simulated cells ordered by their pseudotime from early (top row) to late (bottom row) time points. The derived time-dependent data is visualized with t-SNE. Each dot represents a simulated single cell. The colors represent the inferred pseudotime.

5.5 NEIGHBORHOOD SELECTION RECOVERS NETWORK MODELS

In neighborhood selection our parameter λ determines the degree of penalization of the estimated coefficients in our model. Thus, it influences the shrinkage of the neighborhood size of a node, and coupled with this the density of the estimated network structure (see Fig. 5.1). Ideally, the resulting network should include interactions that contain the most important interactions and should omit interactions that are unnecessary (e.g. indirect associations).

5.5.1 Model selection

The choice of the hyperparameter λ clearly affects the inferred network structure. For this reason a careful selection of λ needs to be made. We introduce a methodology on how to select an appropriate λ that depends on two criteria: Firstly, the cross validation (CV) error and secondly, the stability of a network model.

The cross-validation error is determined by the prediction error between the train and test set during k-fold cross validation procedure. Here, the sample set of the data matrix is split into k partitions. In each iteration we use the $k-1$ partitions to train our model with and use the left out kth partition (test set) to predict its value. The difference between the predicted error and the actual value of the edge weights are reported and averaged across the iterations. In our case we use a 10-fold cross validation. Ideally, the error should be low.

The stability index indicates the agreement of predicted interactions in a sub-sampled data matrix across k runs. In each run we remove a fraction f of the data samples and apply neighborhood selection on the remaining data matrix. We report the predicted interactions and calculate the number of agreeing interactions across all runs for each interaction. We sum up interactions that were reported in each run and divide it by the median number of reported interactions. By this, we normalize for the density of the network which varies with different values of λ. The higher the index the more consistent and reliable the results are. In our realization we remove a fraction (randomly sampled) of 20% and use 10 runs to determine the stability index.

Given any fixed λ associated with a network we calculate the two indices and choose λ appropriately. Thereby, we consider the above mentioned conditions associated with each criterion. With regard to the CV error λ, we determine the λ value associated with the CV error with the maximum decrease (inflection point) and select a range of λ scores around the inflection point. With regard to the stability index we identify λ values that have a high stability score, here > 0.8. We determine overlapping regions of both indices and suggest a range of λ values, that can be used for model selection. Dependent on the choice of λ the sparsity of the network gets chosen accordingly. This two-fold model selection criteria ensures an appropriate trade-off between the CV error and the stability of the reported networks. Figure 5.4 shows the two criteria dependent on λ for model selection. Here, we use an example gene expression dataset. For the first λ values we observe an increase of the CV error associated with low stability scores. Here, the predicted network results in many interactions (not shown) susceptible to changes during the cross validation and stability score calculation. At λ index 6 it reaches its maximum CV error value and decreases monotonously while the stability scores increase reaching a steady value of 1. Note, that the CV error can still decrease while the stability of the network models stays constant as the weight of the predicted edges are considered during cross validation. The last λ corresponding to a CV error of zero results in a network with no interactions. Hence, we neither want to have a full and unstable network (first λ values) nor a stable empty network (last λ values). For model selection, we search for a range of λ values corresponding to a decreasing CV error and a high stability score. For the first case, we identify a range around the "inflection point", here the associated λ value with the maximum decrease in the CV error. For the latter case we filter for those λ values associated to a stability score of greater than 0.8. According to our selection

criteria we look for the intersecting λ range and suggest λ values depicted as a gray shaded region in Fig. 5.4. The final λ value can then be chosen manually by the user for network reconstruction.

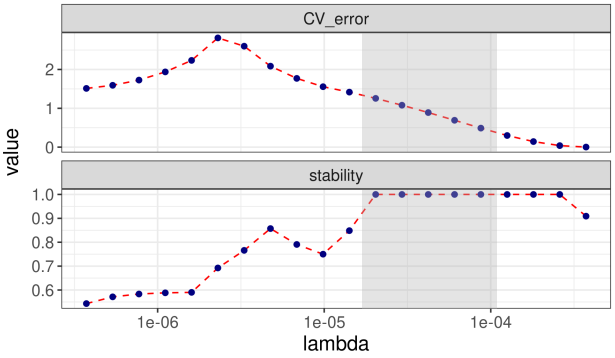


Figure 5.4: Model selection criteria with varying λ.
For model selection we use two different criteria: (i) the cross validation (CV) error and (ii) the stability index. We apply neighborhood selection with 20 different lambda values on an example dataset and derive the two scores. The gray shaded region represent a range of λ values considered as appropriately for model selection. It is determined by a decreasing CV error and high stability score (> 80 %). The x-axis is on log-scale.

Using the two criteria we apply neighborhood selection on a range of different datasets. First, we perform network reconstruction on data generated *in silico* . By introducing dropouts we investigate the effect of different dropout rates on the performance of network reconstruction. We further test our method on time-dependent networks that change dynamically along the simulated trajectories. Moreover, we compare our method with other state-of-the-art methods.

5.5.2 Network prediction performance on *in silico* data

As a proof of concept we use the simulated data reproducing common scenarios in cell biology: the linear developmental trajectory, the cyclic trajectory in periodic cell systems and the multifurcating trajectory branching into three different cell states. Each simulated dataset consists of 2,000 cells with the respective number of genes of the underlying reference networks. For each scenario we estimate λ individually using the proposed criteria (Fig. 5.5, upper panel), and plot the resulting reconstructed gene regulatory networks (Fig. 5.5).

We derive the area under receiver operating characteristic curve (AUROC) in order to compare true network model with the predicted model. The AUROC score is a measure to evaluate the performance of a prediction model and considers the true positive rate (sensitivity) against the false positive rate (1-specificity). AUROC scores of lower than 0.5 are considered as a random prediction model. While evaluating the method we do not consider feedback loops nor the type of interaction. As the neighborhood selection procedure does not consider causal inferences we also omit the direction of the regulation. As mentioned earlier the gray shaded region indicates a suggestion where the optimal λ might be located and is characterized by an area of maximal decrease in CV error values and a considerably high stability index of greater than 0.8.

In Figure 5.5 the straight line represents the λ value located at the inflection point. Using this value we derive the predicted network and report the corresponding AUROC value. Using these λ values located at the inflection points of each of the corresponding trajectory scenarios we obtain AUROC values of 1 (linear), 0.67 (cycle) and 0.57 (trifurcation), ranging from perfect prediction to

an almost random prediction model. However, if we choose other λ values (depicted as a dashed line in Fig. 5.5) we see better performances of AUROC scores of 0.89 (cycle) and 0.66 (trifurcation). Hence, varying λ values in the gray shaded region can lead to better improvements of the network predictions leading to very good to reasonably good performances.

Generally, we observe a gradient in the prediction performance depending on the complexity of the underlying network model. While in the linear scenario we only have perfectly predicted interactions the number of false positives or false negatives increases in the cyclic scenario. However, in the most complex scenario simulating a trifurcating trajectory we see more false positive interactions, as well as false negatives interactions. This drop in performance might be explained by the fact that we neither can resolve self-loops nor have any directional information on the edges. Both types of interactions are crucial in this scenario and cannot be resolved by the neighborhood selection procedure. Hence, the AUROC score is relatively poor compared to the linear and cyclic scenarios. It is worth to mention (although it has not been discriminated in the evaluation) that in majority of predicted interactions the type of interaction agrees with the underlying network model.
Thus, using these synthetic network examples we can confirm that neighborhood selection coupled with the model selection methodology works well on data generated *in silico* .

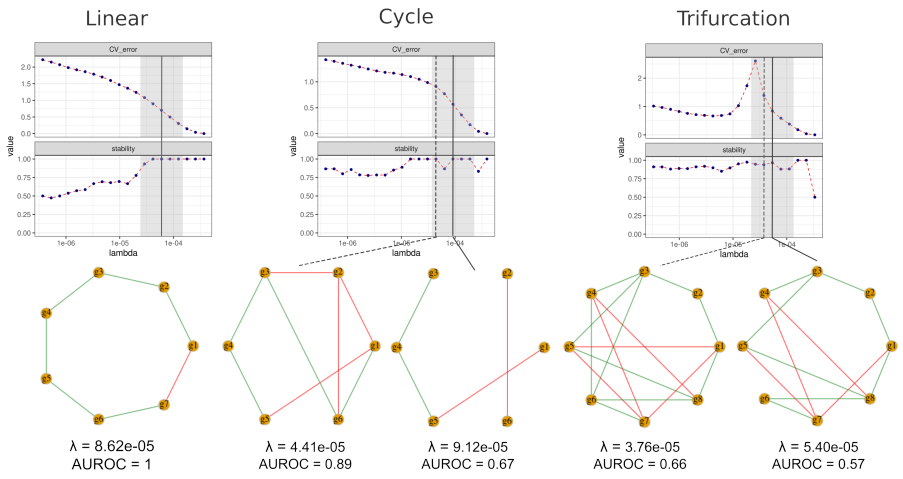


Figure 5.5: Predicted network models by neighborhood selection using model selection criteria.
Using the data generating the linear, cycle and trifurcating trajectories *in silico* we perform neighborhood selection using the model selection criteria (upper panel). Similar to Fig. 5.4 the criteria consist of the CV error and the stability index. We highlight λ values around the inflection point with decreasing CV error rates as well as high stability values. The straight line represents the inflection points whereas the dashed line represent alternative λ values that can be used for network prediction. We plot the predicted networks (lower panel) with respect to the corresponding λ values and report the respective AUROC scores.

Effect of dropout introduction

The data generated *in silico* provides a good framework to test the neighborhood selection method in a controlled environment. However, one drawback of this dataset is the absence of dropouts. Dropouts are commonly seen in single-cell transcriptome data and pose a great challenge during single-cell data analysis. In order to measure the effect of dropouts on the network prediction performance we introduce different dropout rate scenarios seen in earlier studies (Chan et al., 2017; Pratapa et al., 2020):

Given a data matrix, a dropout percentile q and a dropout rate of r we determine for each gene its qth percentile expression level denoted as x_q. Using this qth percentile expression level as a threshold we set expression levels smaller than this threshold with a probability of r to zero. As an example, we set q=50 and r=70 and determine the expression level at the 50th percentile x_{50} for each gene. A gene with a lower expression value of x_{50} has a 70% chance to become a zero and is considered to be a dropout. Fig. 5.6A illustrates the effect of dropout introduction in an example gene. The blue dots represent the original gene expression values. Red dots represent the gene expression values after introducing dropouts with a percentile of 50 and dropout rate of 70%. We test three different scenarios with a dropout percentile of 25, 50, 50 and dropout rate of 50, 50, 70, respectively. Overall, the dropout percentage of the corresponding datasets in our simulation scenarios result in a dropout percentage of approximately 5%, 12% and 17% (Fig. 5.6B). Note, in the original data we do not have any dropouts which results in non-visible bars.

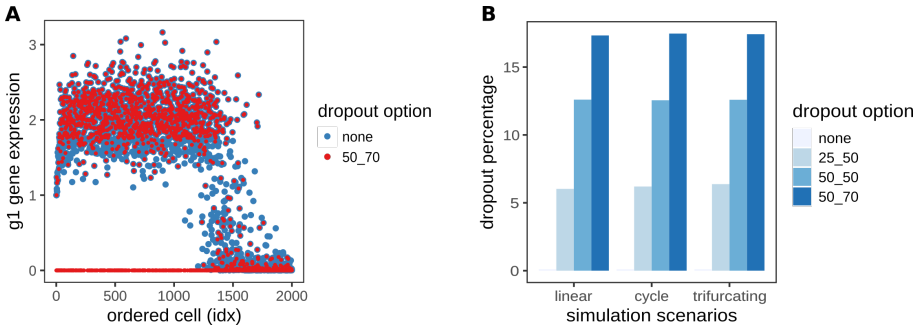


Figure 5.6: Introducing dropouts to simulated data.
(A) Gene expression profile before (blue dots) and after (red dots) introducing dropouts using a 50th percentile and dropout probability of 70%. (B) Overall dropout percentages per dataset using various dropout options.

Raising the number of dropouts disrupts the trajectory profile. After introducing different dropout options to the dataset we inspect the visualization of the trajectory profiles in the t-SNE embedding. Figure 5.7 shows the progression of the linear trajectory before and after dropout introduction using a 25th percentile and a 50th percentile with a respective 50% dropout probability. In the trajectory without any dropouts we see a linear progression over time with a small split in the very early time-ordered cells. Introducing dropouts with a 25th percentile and a 50% dropout probability enhance the difference between those early time-ordered cells and additionally splits up late time-ordered cells from the remaining cell population. Changing the 25th percentile to the 50th percentile disrupts the trajectory profile entirely such that the linear progression is not

visible anymore. Although the overall dropout percentage per dataset is comparably low with respect to real experimental datasets it already has a high impact on the visualization of the trajectory. Thus, we are keen to see what effect the presence of dropouts has on the network reconstruction performance as a next step.

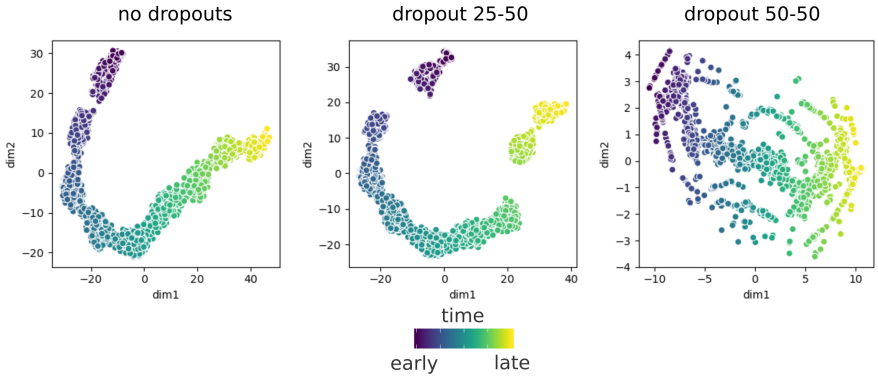


Figure 5.7: Dropouts disrupt visualization of the linear trajectory.
Dataset simulating linear trajectory without dropout introduction (left) and with dropout introduction using 25th percentile and 50% dropout probability (middle) as well as 50th percentile and 50% dropout probability (right). For visualization we applied t-SNE and colored the cells according to their pseudo-time.

Mild influence of dropouts on the performance of network prediction. In order to investigate the effect of dropouts we use the three simulated scenarios and applied the dropout procedure with different dropout percentiles and probabilities. Using the original dataset without dropouts and the respective datasets with varying dropout options we apply neighborhood selection and measured the AUROC scores to evaluate the performances. We use the whole parameter search space of λ and derived the corresponding AUROC scores plotted as a boxplot for each dataset (Fig. 5.8). In the linear scenario we observe a stably very good performance prediction irrespective of the increase of dropout percentages in the datasets. In the cyclic scenario we see similar performances without dropout and with low dropout percentages. Adding more dropouts by using higher percentiles and/ or dropout rates we see a small decrease in performance but still in a range of well predicted networks. Interestingly, in the trifurcating scenario we observe in general better performances scores reaching up to ~0.8 with increasing dropout percentages. Thus, we conclude that in general introducing dropouts to our simulated data has only a minor effect on the network reconstruction. In some cases it can even lead to better performances than without any dropouts.

In summary, we have shown that the neighborhood selection method using the two-fold model selection criteria works well on various types of datasets simulating different developmental trajectories. We have introduced varying dropout rates to increase the dropout percentages in each dataset. Although the embedding of the trajectory gets disrupted with increasing amount of dropouts the performance of network reconstruction using neighborhood selection is only slightly affected. As a conclusion, we confirm that our method is able to recover the underlying reference networks in an adequate way.

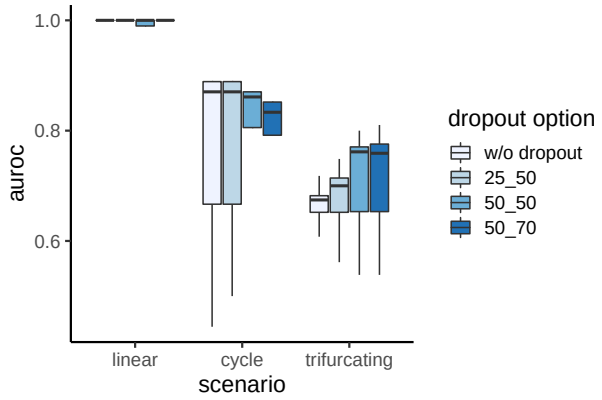


Figure 5.8: Network prediction performance with varying dropout options. Boxplots showing performance scores derived from the whole hyperparameter search space of λ for each simulated trajectory with varying dropout options. Outlier AUROC scores are not shown.

5.5.3 Evolving gene regulatory networks upon time-ordered cells

Single-cell transcriptome technologies allow for profiling thousands of cells that follow developmental processes. Using algorithms inferring the pseudotemporal order of the cells we predict the direction of the trajectories that the cells may follow. In contrast to bulk RNA-seq experiments profiling only a few samples resulting in a lower resolution of sampled time-point experiments it was difficult to infer such a temporal ordering. Hence, single-cell experiments provide us to track developmental paths with a high resolution such that we are able to examine the analyzed system as a dynamic rather than a static biological system.

In this section we want to investigate if we can dynamically reconstruct gene regulatory networks that evolve along a time-ordered path. To test this we designed a concept illustrated in Fig. 5.9. Here, we specify a window size w and infer gene regulatory networks from cells assigned to that window. We denote those networks as ***local*** networks. Then, we move the window with a step size s and continue with the local network prediction until the last window referred to the latest temporally ordered cell population is reached. Finally, we merge all local networks together by aggregating the weights by their average in order to get a globally ***merged*** network.

Evolving networks on a linear trajectory recover reference network

In order to test the concept of dynamically evolving networks we apply the proposed method on an example dataset simulating the linear trajectory. To this end, we sort the cells by their pseudotemporal order by applying `Slingshot` (Street et al., 2018). For illustration issues we use a window size of 1,000 cells and a step size of 500 resulting in a partition of three sub-populations (Fig. 5.10 A+B). However, we apply this method also on a smaller window size of 500 and a step size of 100 cells resulting into sixteen sub-populations making it more challenging for illustrations but the result remains the same for both parameter settings. For each partition we apply neighborhood selection using the same λ parameter estimated by the whole dataset before (λ = 8.62e-05) resulting in locally inferred networks (Fig. 5.10C). These locally inferred networks resolve the time-dependent partition reflecting the gene regulations within the matching time window. Hence, by partitioning the time-dependent data we can observe which part of the gene network is active and which regulations happen in that specific time frame. Interestingly,

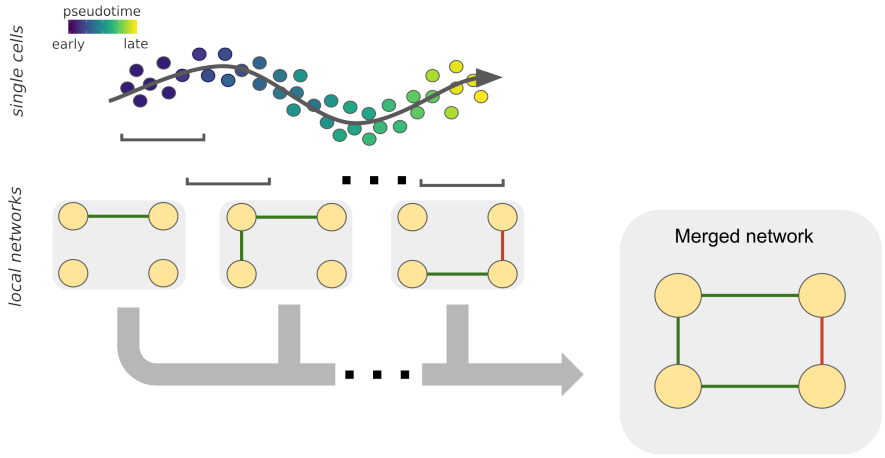


Figure 5.9: Concept of evolving network reconstruction.
Single cells are temporally ordered by pseudotime. We determine a window of size w and infer local networks from single cells assigned to that window. Moving the window by a step size s we continue reconstructing networks until the last window is reached. We aggregate all local networks to get a globally merged network.

by merging the local networks we are able to reconstruct the full reference network similar to when we use the whole dataset before (AUROC=1). Hence, we do not suffer from any lost of information if we first predict local networks and subsequently merge them to a global network.

Local network prediction on cyclic and trifurcating trajectories

Similarly to the linear trajectory scenario we performed local network reconstruction on the cyclic trajectory. Again we sorted the cells by `Slingshot` and use a window size of 500 as well as a step size of 100 cells. Merging the local networks results in a network with an AUROC score of 0.78 displayed in Fig. 5.11A. Using the λ=9.12e-05 at the inflection point we are able to improve the prediction results from 0.67 to 0.78. Taking the alternative choice with λ=4.41e-05 dropped the performance from AUROC=0.89 in the global scale to AUROC=0.81 in the local scale (not shown). Also in the trifurcating scenario we obtain striking results. Here, we separated the dataset into the three lineage branches identified by pseudotime analysis. For each lineage we locally reconstruct networks using λ=5.40e-05 (Fig. 5.11B). The activating interactions between g1 – g2 – g3 re-occur in every locally inferred network and probably refer to the common branch before the lineages diverge. Merging the the local networks results in a network with an AUROC score of 0.71 (Fig. 5.11C). Thus, we observe a performance improvement from almost random (AUROC = 0.57) to reasonably good (AUROC=0.71).

Notably, there are two ways of inferring networks locally: Either we define a certain window size and infer local networks from that partitioned sub-population and move the window forward, or we identify lineage branches and recover lineage-specific networks. Comparing the global network inference as it has been performed in the first part of the chapter to the local network inference performed here we might see superior performances in the local network inference.

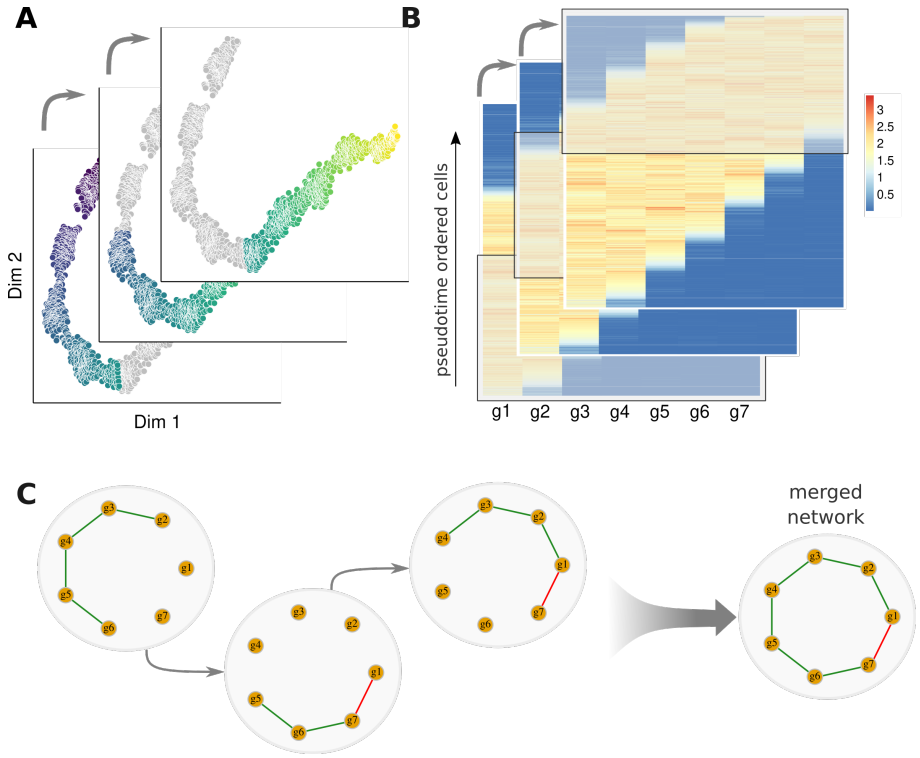


Figure 5.10: Gene regulatory networks evolve along the linear trajectory. (A) Three time-dependent partitions using a window size of 1,000 cells along the linear trajectory. (B) Heatmap showing gene expression data ($g1$,...,$g7$) derived from the respective partitions. (C) Predicted networks using neighborhood selection on each partitioned time-ordered data. The networks partially overlap and recapitulate the respective development over time. The merged network recovers the reference network.

Nonetheless, it remains challenging to reconstruct networks from biological systems with complex interactions which can be seen in the trifurcating scenario. A huge advantage of the local network reconstruction is that it enables us to follow on-going regulations in a specific subset of cells. This leads to a change of perspective in analyzing gene regulatory networks. Instead of reconstructing a static network on a global scale this procedure allows us to investigate the dynamics of a given gene regulatory network in a time-dependent manner.

Influence of window size on network prediction performance

In our procedure the window size w is a parameter that sets the time-frame range we currently look at. Thus, it is a crucial parameter influencing the number of samples used to reconstruct the gene networks. Setting the window size w too small lead to smaller sample sizes such that associations between genes cannot be appropriately estimated (i.e. bulk RNA-seq samples) whereas large window sizes converge towards similar performances related to the network reconstruction on a global scale. We believe that the influence of the step size parameter can be neglected as long as the step size is smaller than the window size such that we obtain overlapping time-frames along the trajectory.

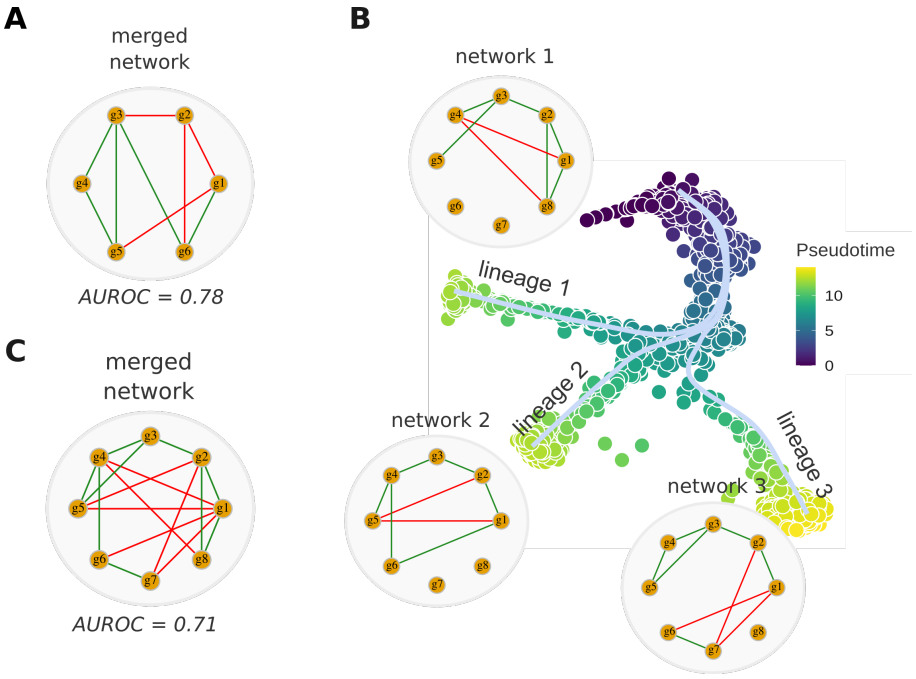


Figure 5.11: Evolving gene regulatory networks along different trajectories. (A) Merged network reconstructed from cyclic trajectory with window size 500 and step size 100 cells using a λ=9.12e-05. (B) Local network inference on trifurcating trajectory by partitioning the data into three lineage-dependent sub-populations using λ=5.40e-05. (C) Merged network by aggregating the local networks from (B) results in in an AUROC score of 0.71.

To test the influence of the window size w we vary the parameter in steps of 100, 250, 500, 750 and 1,000 cells using step sizes of 50 with the first window size and 100 otherwise. We perform analyses on the linear and cyclic trajectory, respectively and report the AUROC scores in Figure 5.12. Notably, the small window sizes of 100 give only poor results in both trajectory scenarios. The performance scores increase with larger window sizes. In the linear case we achieve constant AUROC scores of 1 with window sizes from 500 cells on. In the cyclic scenario the maximum AUROC scores is reached at window size 750 and slightly drops at 1,000 cells. We hypothesize that this trend can be traced back due to the sampling problem. Despite of the the facts that our simulated data consists of less than 10 genes, with relatively simple dependencies, neither implying factors of technical noises nor containing other sources of biases it requires a high amount of samples (more than ten-fold of the number of features) to infer correct associations between those genes. This raises the concerns that even for more complex biological systems is poses a great challenge to investigate gene regulatory networks on a larger scale with high numbers of genes.

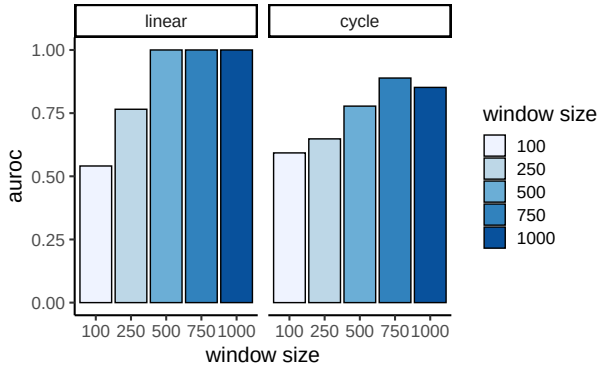


Figure 5.12: Network prediction performance with varying window sizes. Barplots showing performance scores in dependence of different window sizes applied on data simulating the linear and cyclic trajectory.

5.5.4 Neighborhood selection in comparison to other GRN methods

Next, we want to compare the neighborhood selection method coupled with the model selection criteria with other state-of-the-art methods and used the BEELINE pipeline developed by Pratapa et al., 2020. We test different methods that are referred to have superior performances according to the benchmark study published with the BEELINE pipeline. The top three performing GRN reconstruction algorithms are GENIE3, GRNBoost2 and PIDC. Additionally we add PPCOR as a partial correlation based method providing a baseline and SINCERITIES as a regression-based algorithm following a similar approach as the neighborhood selection method.

We report AUROC scores for each trajectory scenario and compared the above mentioned algorithms with (i) our baseline method using λ at the inflection point with a maximal decrease in CV error and high stability index, denoted as 'NB_SELECT', (ii) the alternative choice of λ providing better results denoted as 'NB_SELECT (alt.)' and (iii) the merged network approach from local network predictions along the trajectory, denoted as 'NB_SELECT (merged local)' (Tab. 5.4). In the linear scenario we see similarly to SINCERITIES perfect network predictions with the neighborhood selection method whereas the other tools are slightly worse in the performance with 0.88 or 0.9. In the cyclic scenario almost all method perform equally well between 0.74 and 0.79 AUROC scores, except for our baseline method with 0.67 and the alternative achieving the best results with 0.89. In the trifurcating – the most challenging – scenario we observe mixed results. While our baseline method performs poorly compared to other methods, the alternative neighborhood selection method is on a similar level as GENIE3, PIDC, PPCOR and SINCERITIES whereas our merged local approach is able to compete with the best performing algorithm GRNBOOST2.

Altogether there is no tool performing best throughout all simulated data scenarios. However, with our approach we are able to compete with other state-of-the-art methods. We see superior performances in our 'alternative' and 'merged local' approach compared to our baseline approach and suggest the user to either inspect manually the inferred networks in the suggested λ region provided by the algorithm or pick λ on a global scale and partition the data into many overlapping time-frames. The latter approach allows the user to examine networks in a more dynamical way along a time-dependent trajectory. However, the size of the window needs to be chosen carefully

as it might influence the performance of the network prediction.

Table 5.4: Comparing network prediction performances across GRN methods. Reported AUROC scores for each method and trajectory scenarios. NB_SELECT refers to our neighborhood selection method using two-fold model selection criteria. alt.: is the alternative parameter selection mentioned in Sec. 5.5.2. merged local: refers to the local network predictions treated above.

Tool	linear	cycle	trifurcation
GENIE3	0.88	0.74	0.64
GRNBOOST2	0.88	0.76	0.71
PIDC	0.9	0.75	0.64
PPCOR	0.9	0.75	0.67
SINCERITIES	1	0.79	0.64
NB_SELECT	1	0.67	0.57
NB_SELECT (alt.)	1	0.89	0.66
NB_SELECT (merged local)	1	0.78	0.71

In conclusion, we developed a network reconstruction method that we tested on several simulated datasets with different trajectory scenarios commonly seen in systems biology. We developed a strategy based on a two-fold model selection criteria in order to choose optimally for the network hyperparameter λ. As a proof-of-concept we apply the approach on the simulated datasets and additionally introduced dropouts. We only see minor effects on network prediction performances. Next, we inferred gene regulatory networks in a more dynamic way by moving a time-dependent window along the trajectory and infer local networks specific to that time window. We see better network prediction performances compared to the baseline approach. Finally, we compare our developed method to other state-of-the-art network reconstruction algorithms and consider our method as a competing approach to the other algorithms.

5.6 NEIGHBORHOOD SELECTION ON BLOOD STEM AND PROGENITOR CELL DIFFERENTIATION

5.6.1 Myeloid differentiation as a model system to study gene regulatory networks

In this section we want to investigate the prediction performance of the proposed methodology on a well-examined biological model system. We chose to study blood stem and progenitor cell differentiation which is an extensively studied research field (Iwasaki et al., 2007; Krumsiek et al., 2011; Laiosa et al., 2006; Nestorowa et al., 2016). The cell lineage pathways is depicted in Fig. 5.13A. A population of hematopoietic stem cells (HSCs) differentiates into multipotent progenitors (MPPs) which diverge either into two intermediate cell stages common myeloid progenitor (CMP) or common lymphoid progenitor (CLP) cells. The latter provides precursor cells for lymphocytes while CMP cells can further branch into the megakaryocyte-erythrocyte progenitor (MEP) lineage from which erythrocytes and megakaryotes derive or into the granulocyte/ macrophage progenitor (GMP) lineage from which monocytes and granulocytes derive. The cell lineage specification starting from common myeloid progenitors (CMPs) to megakaryocytes, erythro-, mono- and granulocytes is called ***myeloid differentiation***. As a model system a lot of research has been performed on myeloid differentiation. Thus, the regulation between the genes has been studied in depth throughout the years (see Review by Iwasaki et al., 2007). Krumsiek et al., 2011 curated the literature and proposed a Boolean network (Fig. 5.13B) that models myeloid differentiation using eleven transcription factors. These lineage-specific transcription factors govern the process of cell fate decision and can be can be classified into three groups (Krumsiek et al., 2011). In parentheses we denote genes name aliases:

1. Early hematopoietic factors
    - Gata2, Cebpa
2. Intermediate factors
    - Gata1, Pu1 ("SPI1")
3. Secondary fate determinants and cofactors
    - Eklf ("KLF1"), Fli1, Fog1 ("ZFPM1"), Scl ("TAL1"), Gfi1, cJun ("JUN"), EgrNab (integration of "EGR1", "EGR2" and "NAB2")

We use the Boolean network proposed by Krumsiek et al., 2011 and simulate gene expression data similarly to the synthetic networks using BoolODE. Figure 5.13C visualizes the cell lineages as well as their inferred pseudotime in a 2D t-SNE representation. Notably, the central cell population represent the earliest cells differentiating into three visible branches. Figure 5.14 shows specific gene expression values for each cell lineage. The middle cell population consist of mainly CMP cells characterized by *Gata2* gene expression. Starting from CMP cell population may emerge (i) to the top right branch representing granulocytes defined by high *Gfi1* expression, (ii) to the top left branch representing monocytes defined by high *EgrNab* expression, or (iii) downwards either

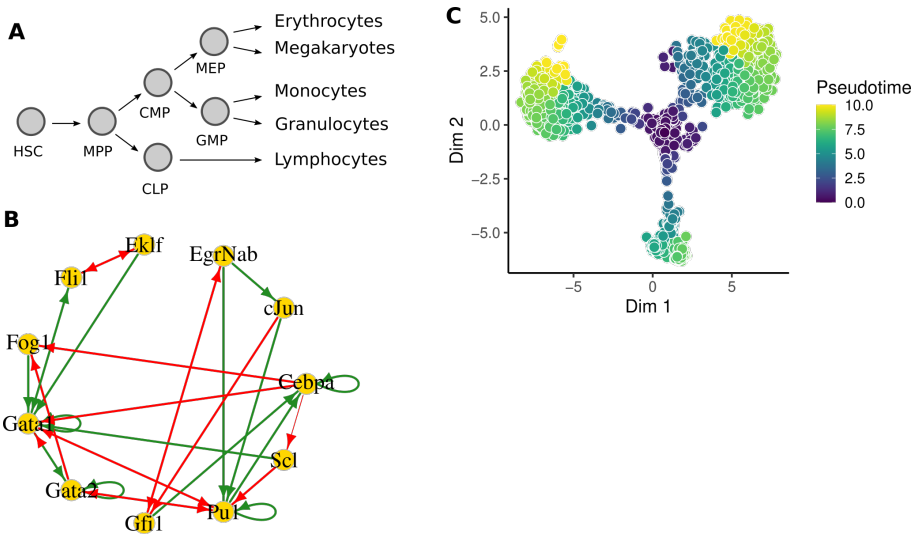


Figure 5.13: Literature-curated model for myeloid differentiation.
(A) Differentiation pathways from hematopoietic stem cells to different blood cell types. HSC: hematopoietic stem cells, MPP: multipotent progenitor, CMP: common myeloid progenitor, CLP: common lymphoid progenitor, MEP: megakaryocyte-erythrocyte progenitor, GMP: granulocyte-monocyte progenitor. (B) Literature-curated gene regulatory network of eleven marker genes controlling cell fate decision into erythrocytes, megakaryotes, monocytes and granulocytes. Subfigures (A) and (B) are modified from Krumsiek et al., 2011. (C) t-SNE visualization of simulated myeloid data with boolODE using the GRN model from (B).

to erythrocytes showing high *Eklf* expression or megakaryotes showing high *Fli1* expression level.

As the pseudotime method Slingshot has been able to only infer three lineages appropriately we decide to combine the erythrocyte and megakaryocyte cell lineage as one cell lineage. Since both cell states derive from the same intermediate cell state (MEP) the combination of the two cell populations is also biologically meaningful.

5.6.2 Network reconstruction on simulated myeloid differentiation data

We use the two-fold model selection criteria in order to estimate the λ hyperparameter for network reconstruction. The algorithm yields a parameter range depicted in the gray box in Figure 5.15 defined by a fast decrease in the cross-validation error and high stability index. We choose a λ value of 0.0001 and inferred local networks for each cell lineage (Fig. 5.16A).

The cell lineage branch emerging towards erythrocytes and megakaryocytes reveal a gene regulatory network in which the cell-type specific lineage factors *Eklf* and *Fli1* controlling for erythrocytes and megakaryocytes respectively are both involved in the regulatory network. The remaining interacting transcription factors (except for *Gfi1*) are known to regulate cell fate commitment from early CMP cells to erythrocytes and megakaryocytes. In the next cell lineage branch

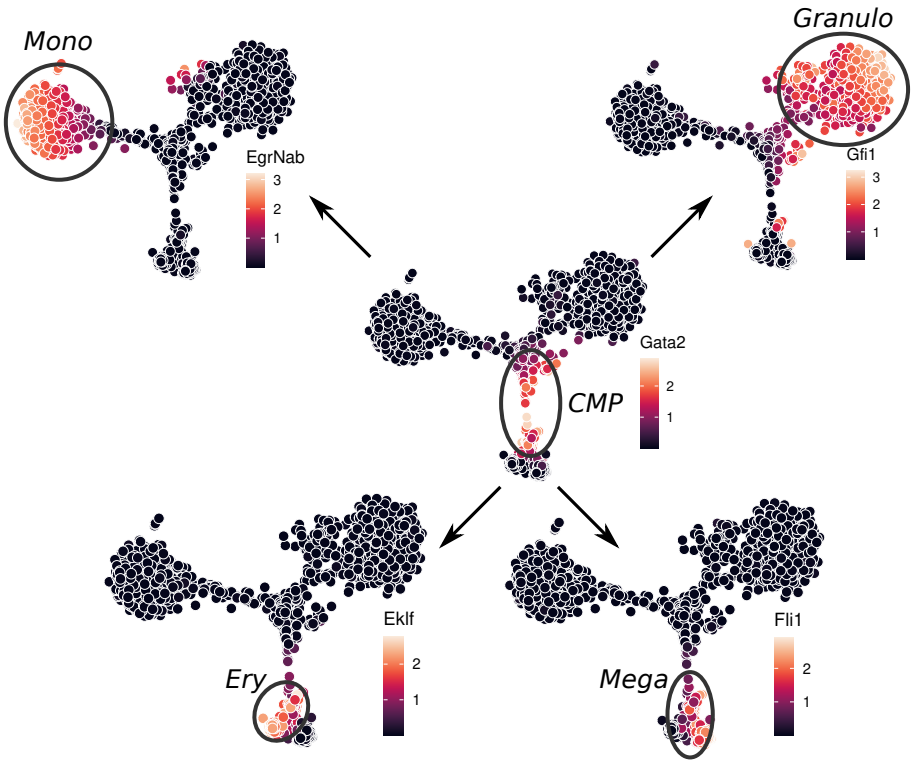


Figure 5.14: t-SNE visualizations of simulated cell lineages during myeloid differentiation.
Common myeloid progenitor (CMP) cells differentiate into monocytes (mono), granulocytes (granulo), erythrocytes (ery) and megakaryotes (mega). Marker genes characterize the cell type identities are color-coded respectively.

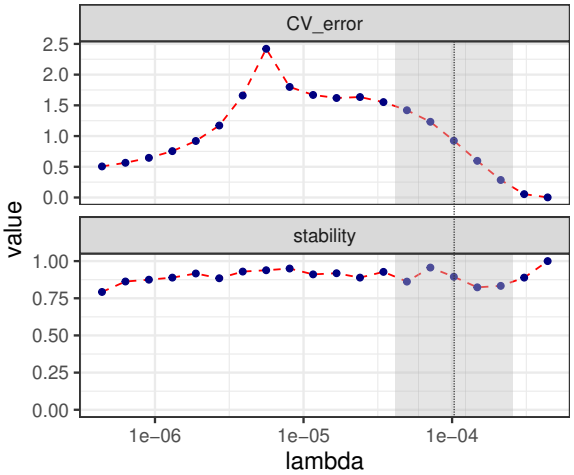


Figure 5.15: Model selection on simulated myeloid differentiation data.
Using the whole simulated dataset on a global scale the algorithm suggests an area (gray shaded box) around the inflection point at λ=0.0001 (dotted line). We use this value for reconstructing the gene regulatory networks shown in Fig. 5.16.

differentiating towards granulocytes we see regulatory interactions involving the lineage-specific factor *Gfi1*. Again, early hematopoietic factors such as *Cebpa*, *Gata2* and *Pu1* are part of the regulatory network reconstructed by our method. In the last cell lineage branch giving rise to monocytes *EgrNab* as the lineage-specific marker is involved in the predicted regulatory network. Remarkably, all regulatory interactions predicted in this cell lineage agree with the underlying literature-curated Boolean network and thus are true positive predictions. Aggregating all local

networks lead us to the merged gene regulatory network plotted in Figure 5.15B with an overall prediction performance of an AUROC=0.65. In comparison to that, the global network reconstruction performs slightly minor with an AUROC=0.61.

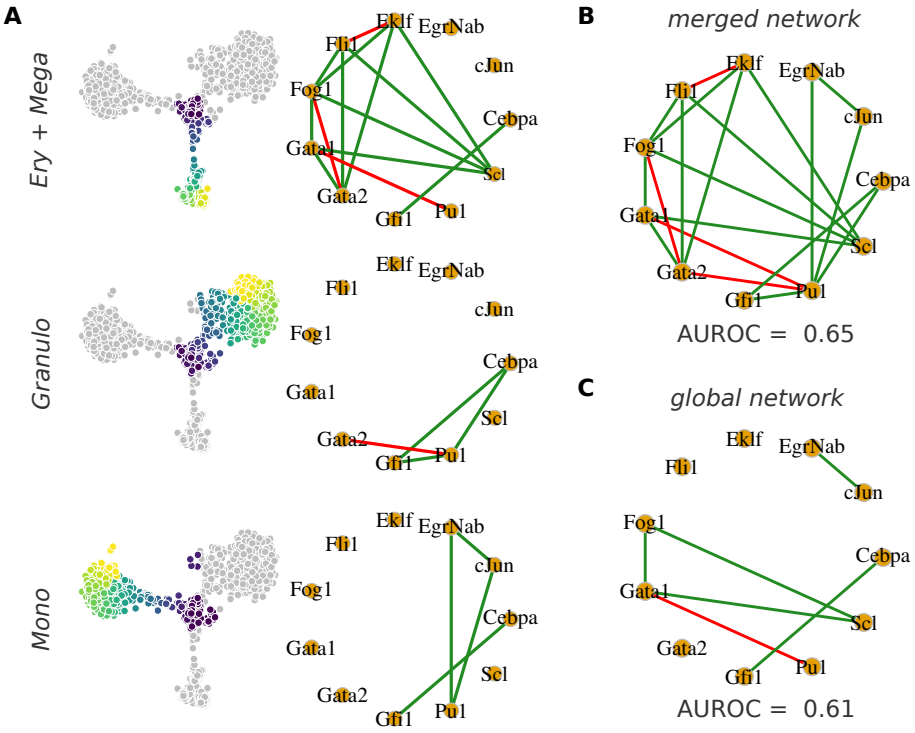


Figure 5.16: Reconstructed gene regulatory networks during myeloid differentiation.
(A) Locally reconstructed GRNs for each cell lineage using $\lambda = 0.0001$. Erythrocytes and megakaryotes are fused indicating the megakaryocyte-erythrocyte progenitor cell type. Cells are color-coded by the inferred pseudotime. (B) Merged GRN from local networks reconstructed in (A) leads to an AUROC value of 0.65. (C) Globally inferred GRN leads to a minor prediction performance of AUROC=0.61.

A clear advantage of the local network inference is not only the better performance but also the possibility to examine specific gene regulations for a particular cell lineage. The above analysis has shown which factors are involved in regulatory interactions at a particular period of time. This allows us to reveal active gene regulations in a lineage-specific manner.

5.6.3 Network reconstruction on experimental hematopoietic stem cell differentiation

Using simulated data with an underlying ground truth dataset provides a useful approach in a controlled environment to test the performance of a computational method. However, oftentimes the simulations do not reflect the full range of computational challenges that one is faced in experimental datasets. In our case, the simulations performed in this chapter neither suffer from sources of technical biases nor the curse of high-dimensionality where the number of features

(i.e. genes) is much higher than the number of samples (i.e. cells). In this section we pursue gene regulatory network reconstruction on an experimental dataset published by Nestorowa et al., 2016. As an analogue of the previous dataset simulating myeloid differentiation Nestorowa et al., 2016 provides a single-cell map in mice of hematopoietic stem and progenitor differentiation emerging towards early stages of myeloid and lymphoid cell types. Lymphoid cells comprise cell types that are part of the innate immune system such as B-cells, T-cells or natural killer cells.

Data preparation. We use the already preprocessed and normalized mouse hematopoietic stem cell (mHSC) data published along with the BEELINE framework (Pratapa et al., 2020) (Fig. 5.17A). After quality control and filtering it constitutes of 1,656 cells and 4,762 genes. As originally performed by Nestorowa et al., 2016 we use diffusion maps to visualize the data in a 3-dimensional representation (Fig. 5.17B). Following Pratapa et al., 2020 we use the data split into three cell-lineages: erythrocytes, granulomonocytes and lymphoid cells (Fig. 5.17C). In order to reduce the dimensions we select the 500 highly variable genes (HVGs) along the pseudotime together with significantly varying transcription factors for each cell-lineage correspondingly. The sizes of respective sub-datasets denoted as mHSC-E (erythrocytes), mHSC-GM (granulomonocytes) and mHSC-L (lymphocytes) are summarized in Table 5.5. Note that common progenitor cells across the lineages can occur several times. For each sub-dataset we now infer GRNs using neighborhood selection.

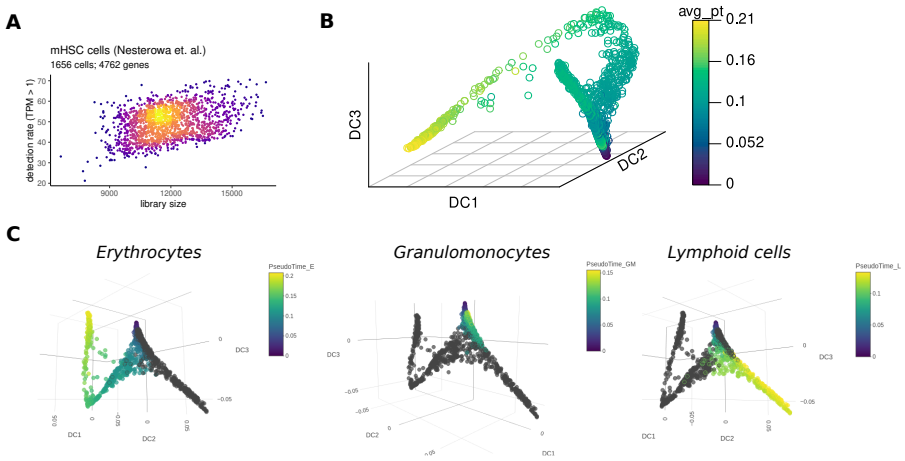


Figure 5.17: Single-cell RNA-seq data of mouse hematopoietic stem cells (mHSCs) by Nestorowa et al., 2016.
(A) Data quality plotting detection rate in dependence of library size. Dots represent single cells and are color-coded by the dots' density. Dataset comprises 1,656 cells and 4,762 genes after quality control. (B) First three diffusion components of mHSCs color coded by pseudotime. (C) Cell lineages with respective pseudotime information of erythrocytes, granulomonocytes and lymphoid cells in a 3D representation using diffusion maps.

Evaluation. As a measure of evaluation we use the early precision ratio (EPR) score and only consider interactions between transcription factors likewise to the BEELINE framework. Most of the network reconstruction algorithms infer networks that are close to a full graph. Being more interested in the most important predicted interactions we analyzed the interactions with

Table 5.5: Data statistics of mHSC data. Overall, 1,656 were analyzed and split into three cell-lineages using pseudotime analysis. Note that some (early progenitor) cells can occur multiple times. We select the top 500 HVGs and significantly varying transcription factors (TFs) along pseudotime.

Cell type	numCells	numGenes+TFs
mHSC-E	1,071	704
mHSC-GM	889	632
mHSC-L	847	560

the highest (absolute) weights and thus the top ranked interactions within the network (top-k network). For this reason and to evaluate the performance of each inferred network based on using early precision (EP) score which is given by the number of true positives divided by the number of positively predicted observations within the top-k network. In order to maintain comparability across datasets we divide the EP score by the network density of each evaluation subnetwork obtaining EP ratios (EPR). Thus, EPR of 1 is indicative of a random predictor consistently. With regard to the evaluation network we use the STRING database (Szklarczyk et al., 2019) and selected for each sub-datasets only the evaluation sub-networks including genes overlapping with the corresponding sub-dataset. For the comparison with the evaluation network, we only consider and filter for edges going out of transcription factors.

Neighborhood selection on mHSC data

We use the two-fold model selection criteria in order to estimate an appropriate λ value (Fig. 5.18A). Across all sub-datasets the curve of the CV error line is similar. It is characterized by an increase of CV error reaching its maximum value at around λ index 6 decreases continuously. While the CV error decreases from λ index 6 the stability score stays very low with respect to mHSC-E and mHSC-GM data. Here, the choice of an optimal λ remains ambiguous as the majority of λ values leads to unstable network predictions. Hence, there is no region suggested by the algorithm in these sub-datasets. In mHSC-L data we see high stability scores around the inflection point of the CV error curve. Here, we can provide a region to the user fulfilling the model selection criteria. Leaving the stability index unattended for now in mHSC-E and mHSC-GM data, we choose to use the λ value with the maximal decrease in the CV error depicted as a dotted line across all three sub-datasets.

We use the respective hyperparameter value and predicted the GRNs correspondingly (Fig. 5.18B). Unfortunately, due to the hairball-like structure the reconstructed networks are hardly readable but should provide a glimpse about the complexity of the predicted networks. For evaluation we inspect the top-k network filtered for transcription factor - gene interactions as stated above (Fig. 5.18C). We calculate EPR scores resulting in decent performances in case of mHSC-E and mHSC-GM achieving EPR scores of 2.87 and 3.94, respectively. However, mHSC-L data is close to the random prediction model with an EPR score close to 1. Using the biological knowledge about erythrocyte and granulo-monocyte differentiation we can further search for interactions in

which lineage-specific factors are involved (Fig. 5.18D). During erythrocyte differentiation *Eklf* (here *KLF1*) is a marker that is part of the top-k network. Indeed, we are also able to reconstruct interactions with *GATA1*. With regard to the granulo-monocyte differentiation *GF1B* and *EGR1* (as part of the *EgrNab* complex) present important factors that we use for filtering within the top-k network. Here, neighborhood selection predicts interactions with *GATA2* which is part of the myeloid differentiation process.

Although mHSC-E and mHSC-GM did not fulfill the model selection criteria due to unstable network predictions, the performance achieve better results in comparison to the mHSC-L data and also lead to predictions with biologically meaningful interactions. The attempt of using a λ index with a higher stability score (at the index around 19) in mHSC-E or mHSC-GM data yield to almost empty networks with random prediction performances (not shown). In contrast to that, in mHSC-L data fulfilling the two-fold model selection criteria leads to random predictions which makes us wonder how the experimental dataset behave and if the model selection criteria are appropriate for sparse and noisy data. We suspect that different data issues such as the number of dropouts or the presence of multiple heterogeneous cell states influences the prediction performances. This shows how complex experimental datasets are and how different the results can turn out in comparison to a simulated framework.

Comparison to other GRN algorithms. Next, we compare the prediction performance of our proposed method to other state-of-the art GRN reconstruction algorithms. We use results obtained from he BEELINE framework and report the EPR scores of the top 5 algorithms in Table 5.6 and added the recorded EPR obtained by our analysis from Fig. 5.18. We observe minor prediction performances in comparison to the top 3 algorithms PIDC, GENIE3 and GRNBoost2 but better results than PPCOR (except for mHSC-L data) and SINCERITIES. While SINCERITIES performed quite well in the synthetic datasets it scores very poorly for the experimental dataset. This reveals that the difference between simulated and experimental datasets can diverge to a great extent even in established algorithms. However, with regard to our computational running time there is still room for improvement. While the analysis lasted only for 1-2 minutes for the synthetic network datasets it took approximately 1-2 days to finish the calculations for the experimental dataset.

Table 5.6: Performance comparison to other GRN reconstruction algorithms. We report EPR scores comparing the prediction performance across state-of-the-art algorithms for network reconstruction. EPR results were adapted from the BEELINE benchmark paper (Main Fig. 5 by Pratapa et al., 2020) and added the EPR scores by our proposed neighborhood selection method.

Cell type	PIDC	GENIE3	GRNBOOST2	PPCOR	SINCERITIES	NEIGHSELECT
mHSC-E	7.49	6.7	6.05	1.56	0.74	2.87
mHSC-GM	8.36	8.7	7.96	1.89	0.4	3.94
mHSC-L	6.22	6.83	6.67	4.33	0.76	1.06

Although our proposed method competes in the middle range of the top5 state-of-the-art GRN algorithms we are still surprised how different the prediction performances between simulated

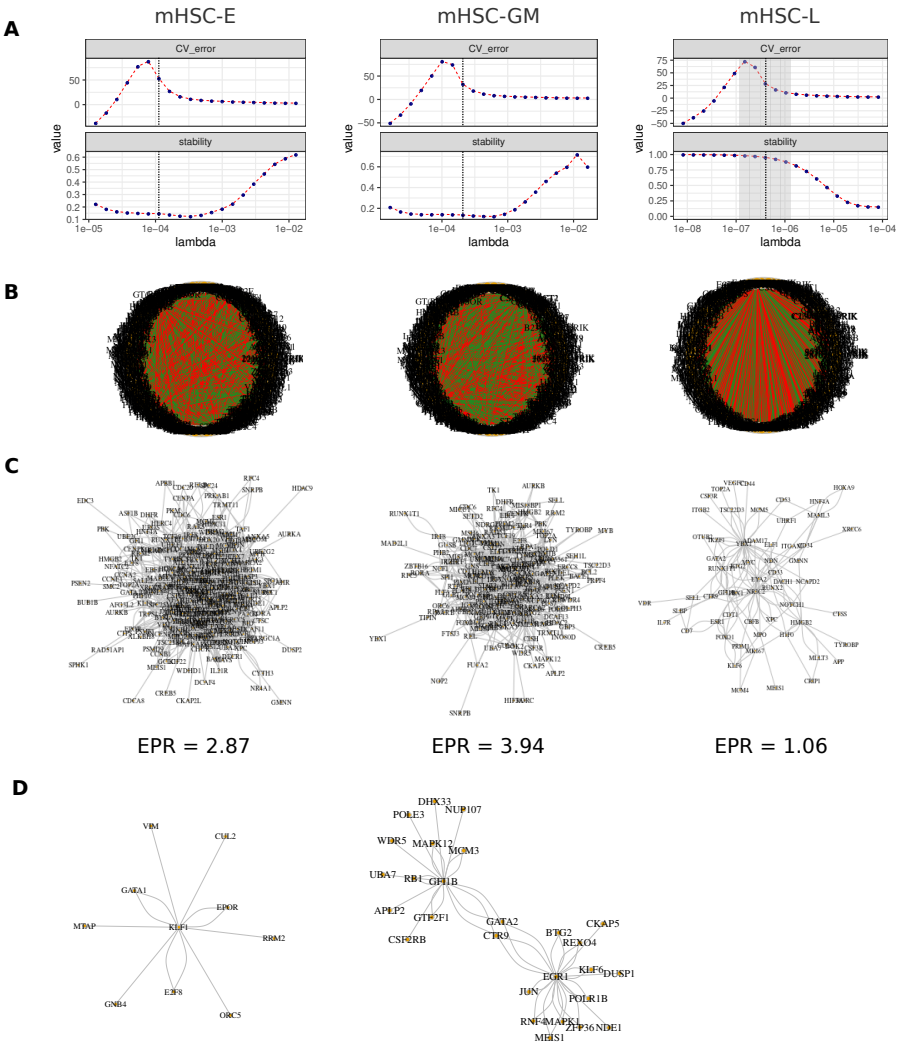


Figure 5.18: Network reconstruction on mHSC data.
(A) Model selection for hyperparameter estimation for each sub-dataset. (B) Reconstructed gene regulatory networks. (C) Top-k networks used to determine early precision ratios. (D) Selected top-k network including lineage-specific transcription factors: *KLF1 (Eklf)* for mHSC-E; *EGR1 (EgrNab)* and *GFI1* for mHSC-GM. No interactions for lineage-specific transcription factors for mHSC-L were found.

and experimental datatsets can emerge. As we suspect the sparsity and the high level of noise might be a source of this divergence we think of imputation methods to counteract against theses challenges. For this reason we aim to apply an imputation method prior to network reconstruction that can possibly improve the prediction results in the next section.

5.6.4 Network reconstruction with data imputation

To date there are many data imputation methods available. As pointed out earlier (Section 3.1.5) imputation methods predict zero counts in single-cell RNA-seq data and can be sometimes used to normalize the data in addition. We address different imputation methods explicitly later in

Section 6.1. As we have shown before, the neighborhood selection method on experimental data performs in a decent way but by far from optimal if we compare it with the other top 3 GRN reconstruction algorithms. We assume that this is caused by the high numbers of zero counts and the noise level present in single-cell RNA-seq data. Therefore, we select an imputation method called `dca` (Eraslan et al., 2019). `dca` stands for a deep count autoencoder and is a deep learning based tool that denoises single-cell RNA-seq data. It claims to facilitate downstream analyses that "enhances the modeling of gene regulatory correlations" (Eraslan et al., 2019). It can be used as an 'all-inclusive' tool reducing the amount of zeros, normalization and dimensionality reduction that can be applied as a preprocessing step before network inference (Eraslan et al., 2019). Intrigued by the usability we apply `dca` to our dataset.

Comparing the gene expression distribution before and after imputation we observe much higher values after imputation (Fig. 5.19). While in the data without imputation we see a high peak of zero counts followed by a bimodal distribution of gene expression values we see no zero counts and a unimodal distribution after imputation using `dca`.

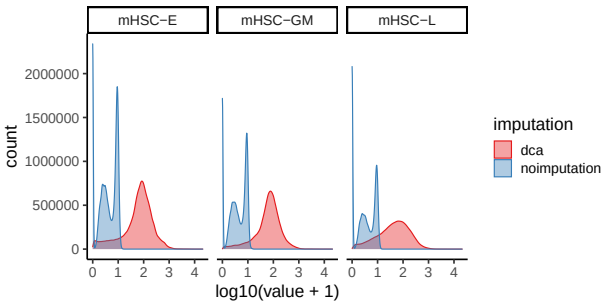


Figure 5.19: Distribution of gene expression values before and after imputation.
Comparison of density plots before (blue) and after (red) imputation using dca. A pseudo count was added to the gene expression values. X-axis is on a log-scale

Next, we investigate how the correlations between genes alter after imputation. Therefore, we select the known myeloid transcription factors that are present in the mHSC-E dataset and plotted pairwise gene-gene relationships (Fig. 5.20). As expected, `dca` overall enhances gene-gene correlations. We observe changes in correlations, e.g. from 0.014 to 0.552 (*EGR1 – GF1*) or 0.433 to 0.852 (*EGR1 – JUN*). These interactions are also reported in the reference network (see Fig. 5.13B). However, there are cases where the correlation values get enhanced but without a literature-curated evidence, e.g. *KLF1* (*Eklf*) – *EGR1* (*EgrNab*) changing from -0.136 to -0.665 or *GATA2* – *JUN* (*cJun*) changing from 0.183 to 0.508. Another example and as an exception, the gene-gene relationship between *SPI1* (*Pu1*) and *GATA2* decreases from 0.11 to -0.0003. Here, we actually expect a higher statistical dependence since the reference model reports an interaction between the two genes.

Intrigued by the changes in correlation values we apply neighborhood selection using the two-fold model selection criteria and evaluate the prediction performance (Fig. 5.21). In contrast to the experimental data without imputation the algorithm is able to suggest a region that fulfills the model selection criteria: A strong decrease in the CV error and a high stability score. We select the λ parameter as usual, depicted as a dotted line in Fig. 5.21A. However, choosing the λ value as stated leads to very sparse network predictions resulting in EPR scores equal to zero across all sub-datasets. We also try to use alternative λ values for network reconstruction and plotted the

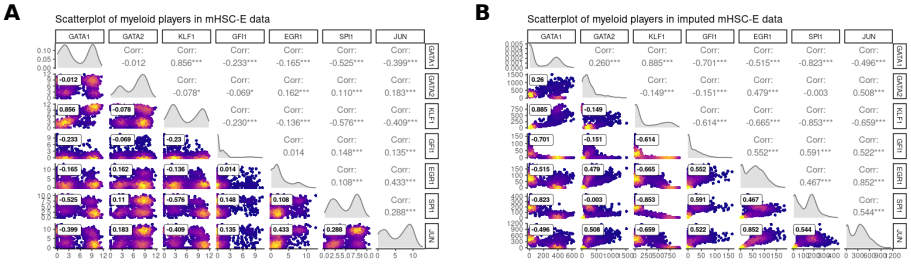


Figure 5.20: Pairwise gene-gene scatterplots in mHSC-E data.
(A) Gene-gene relationships of selected myeloid genes before imputation on pre-processed and normalized mHSC-E data. Every dot represents a single cell, colors represent the density of the dots. Gene names are specified as their aliases (see Sec. 5.6.1). Correlation values are measured in Pearson's correlation coefficient. (B) Gene-gene relationships after imputation using dca.

results in Supp. Fig. A1. Even though the structure of the predicted networks is denser compared to Fig. 5.21B the overlap between the alternative reconstructed GRNs and the evaluation networks is still very little resulting in poor performances. Hence, we do not see any improvement of network inference using `dca` and the proposed neighborhood selection method.

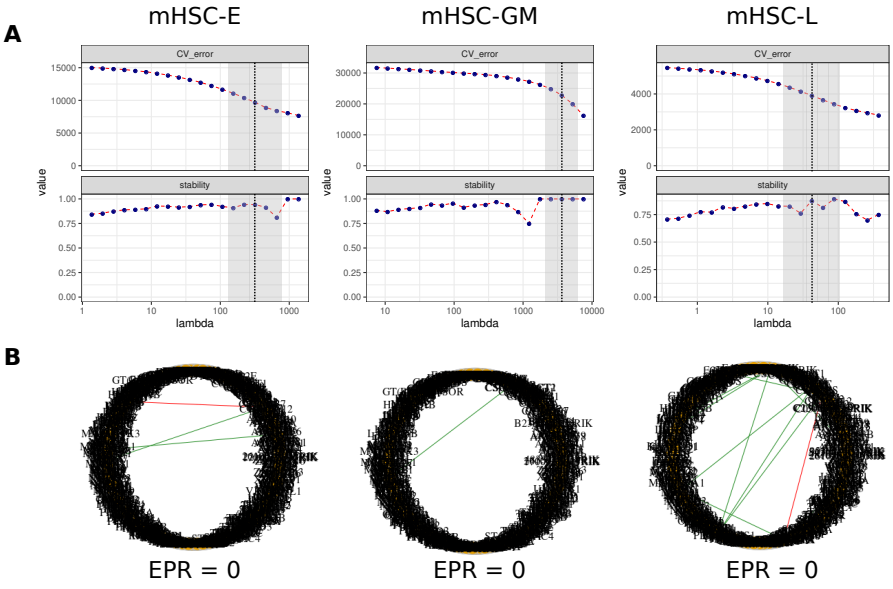


Figure 5.21: Data imputation and GRN reconstruction on mHSC data.
(A) Model selection criteria on imputed data. Across all datasets the algorithm is able to suggest a region fulfilling the conditions for parameter estimation. (B) Reconstructed networks using the λ values depicted as a dotted line in (A). Prediction performance is very poor with EPR=0 across all datasets.

5.7 DISCUSSION AND CONCLUSION

We have introduced a network reconstruction method using neighborhood selection with a hyperparameter estimation. The hyperparameter is selected based on two-fold selection criteria. Using the cross validation error and the stability index of the reconstructed networks we have proposed a strategy to choose the hyperparameter λ. Here, the algorithm suggests a range of λ values fulfilling two conditions: (i) λ values associated with a strong decline (around the inflection point) regarding the cross-validation error and (ii) λ values associated with a high stability index. On the one hand, the CV error condition ensures that the prediction error during 10-fold cross validation gets smaller and thus network edge weight predictions associated with the specific λ are more reliable. On the other hand, the stability index ensures stable network predictions with respect to the reported interactions across multiple runs using a sub-sampling procedure. Thus, the stability index indicates how consistent the network predictions are. Together the two criteria allow us to give us some confidence about the choice of the hyperparameter λ.

We have shown the usability of our method on different datasets. This first includes data generated *in silico* from known network models that were used to simulate three common trajectories in single-cell transcriptomics. We have further investigated the network inference using neighborhood selection based on time-ordered data and observe the possibility to study time-evolving networks as well as lineage-specific networks. Lastly, we use our approach to infer gene regulatory networks on a well studied biological model system with both simulated data and experimental single-cell RNA-seq data examining blood stem and progenitor differentiation in mice.

The first simulation dataset provides clear trajectory scenarios to evaluate network models in a controlled framework. We use that dataset to verify the validity of the neighborhood selection procedure. With increased network complexity characterized by larger number of edges, we have observed difficulties in predicting the underlying network structure. However, indirect edges due to transitive associations could be successfully resolved using neighborhood selection. Furthermore, we introduced dropouts as an attempt to approximate the datasets generated *in silico* to a more single-cell like dataset. We see only mild effects on the network prediction performance by introducing dropouts. However, the data misses many other characteristics of single-cell data such as the high number of features which oftentimes exceed the number of samples as well as other sources of noises and biases.

Inferring networks from time-dependent data is an active area of research. We have used synthetic trajectory scenarios and have been able to predict time-dependent gene regulatory networks that evolve through time-course data along a linear trajectory. By this, it has become possible to predict gene-regulatory interactions that are particularly active in a corresponding time frame and examine the change of network structures along time. Using this approach gene regulatory networks can be even inferred from complex trajectories by first dividing multifurcating trajectories to multiple linear trajectories, separately inferring local networks, and subsequently assembling them to form a global network. We observe superior network prediction performances than reconstructing GRNs on a global scale. However, integrating pseudotime analysis to order the cells in a temporal axis with gene regulatory network reconstruction requires the cells to be correctly ordered over time and hence might pose the risk of being sensitive with respect to

the pseudotime algorithm. Nevertheless, we are convinced that in the future the development of network inference algorithms will consider temporal information in single-cell RNA-seq data in order to examine dynamical changes along one or multiple trajectories.

The most challenging task was to predict gene regulatory networks in experimental data. The chosen experimental dataset is a well-studied model of cellular differentiation with literature-curated marker genes and their regulation. We have used a workflow for network reconstruction previously applied by Pratapa et al., 2020 and filtered the datasets by the 500 highest variable genes and significantly varying transcription factors along the pseudotime. We applied the neighborhood selection method and evaluated the predicted networks following the protocol published by Pratapa et al., 2020. By this, we establish comparability between the benchmark paper of Pratapa et al., 2020 to our proposed method. Although the conditions of our the two-fold model selection criteria could not be fulfilled for two of three datasets, the predicted networks are reasonable and have decent prediction performances that are able to compete with state-of-the-art GRN algorithms. Still, the divergence between the good performance prediction on simulated data and the decent performance on experimental datasets make us think about possible reasons for this observations. We suspected the high amount of zeros causing high levels of noise and thus obscuring biological heterogeneity are the reasons that the algorithm has difficulties in predicting associations among genes.

To remedy this we apply an imputation method that predicts zero counts and denoises the data simultaneously. However, the results with imputed data were very poor leading to very sparse network results. Hence, we wonder what effect imputation methods have on network reconstruction in general and if it is even suitable to infer gene regulations from imputed data.

Lastly, we want to address the choice with regard to hyperparameter λ. The network results and its performance of the network reconstruction are highly dependent on the selection of λ. While in the linear trajectory scenario the model selection of λ is stable across a stretch of λ value ranges the choice of λ does make a strong difference in other scenarios (see Fig. 5.8, w/o dropout). A proper sensitivity analysis on a variety of λ value is missing and can be resolved in future work. However, our observation is that the more complex the data and the underlying network model is, the more sensitive is the resulting network model towards the selection of λ.

As a concluding remark, this chapter provides multiple approaches tested on several datasets to reconstruct networks using neighborhood selection with a two-fold strategy for parameter optimization. The approach is able to compete with current state-of-the-art network reconstruction algorithm. However, as we observe a strong distortion of the reconstructed network using dca imputation we suspect that imputation does not improve network reconstruction in single-cell RNA-seq data, and intend to extensively investigate the relation between imputation and network reconstruction in a systematic way in the next chapter.

6 EFFECT OF IMPUTATION ON GENE REGULATORY NETWORK RECONSTRUCTION

A single-cell RNA-seq pipeline can be built up with many different preprocessing and analyzing tools (Luecken et al., 2019; Vieth et al., 2019). In Chapter 3 we introduced some preprocessing modules that can be combined successively in a data analysis pipeline. They include normalization, bias correction, imputation and feature selection. Preprocessing the data has a crucial effect on the results obtained from downstream analyses: For example, omitting normalization techniques that include variance stabilization results in the undesirable detection of lowly expressed genes with high variances when performing a differential expression analysis. Or, not correcting for systematic biases results in the detection of unwanted confounding factors. In these cases omitting the preprocessing steps may distort the results. Certainly, it matters which tools in which context have been used. Previous studies have systematically evaluated different combinations of preprocessing modules on the result of various analyses steps (Hou et al., 2020; Vieth et al., 2019). Vieth et al., 2019 studied the interplay between different combinations of modular tools and their effect of differential expression analysis. Hou et al., 2020 benchmarked imputation methods with regard to differential expression analysis, clustering and visualization as well as pseudotime analysis. However, it still remains unclear how preprocessing affects network results obtained by network reconstruction algorithms (Blencowe et al., 2019).
As we observed in the previous chapter the imputation of single-cell RNA-seq data resulted in a drop of performance when reconstructing gene regulatory networks with neighborhood selection. Hence, we are particularly interested how the inclusion of a preprocessing module, here imputation, affects the performance of network reconstruction in general. To study this question, we systematically evaluated different combinations of imputation methods and GRN reconstruction methods using multiple experimental datasets. We build upon previously published benchmark studies and select the best-performing computational tools for both imputation and network reconstruction (Hou et al., 2020; Pratapa et al., 2020).

The chapter is based on an article published in *Patterns* (Ly et al., 2021). It is structured by first, an overview of current imputation methods; second, the impact of imputation on gene-gene relationships; third, the systematic evaluation of network results obtained from imputed data; fourth, the analysis of obtained network models with regard to network similarities and motif analysis and finally a discussion. The article can be read under this link: `https://doi.org/10.1016/j.patter.2021.100414`

6.1 STATE-OF-THE-ART IMPUTATION METHODS

As a common preprocessing step imputation methods are used to predict missing values and to smooth the data in single-cell RNA-seq experiments. This reduces the noise level and is ought

to facilitate downstream analyses. In contrast to normalization which corrects for different read depths between cells ensuring comparability across samples and genes, imputation methods estimate unobserved read counts in cases where the method deems that experimental or technical noise has led to the absence of a count (dropouts). In some imputation tools a normalization step is integrated while in other tools imputation can be applied on previously normalized data.

Up to date, a wide range of imputation methods are publicly available (Chen et al., 2018a; Dijk et al., 2018; Eraslan et al., 2019; Huang et al., 2018; Linderman et al., 2018; Lopez et al., 2018; Mongia et al., 2019; Tang et al., 2020; Wagner et al., 2017). The models that the methods are based on underlie different assumptions on the data distributions and can predict linear or non-linear gene-gene relations. We present a few state-of-the-art methods for data imputation that are categorized into four three categories: model-based, smoothing-based and deep-learning-based algorithms (Hou et al., 2020).

Derived from Hou et al., 2020, Table 6.1 provides an overview of selected available methods and their underlying concept with their respective assumptions on the data distributions. Although the list does not provide a complete list of publicly available tools it provides a rough overview about state-of-the-art methods and their underlying methodology. Other tools that are not listed have a similar mathematical concept implemented. Below, we further describe briefly how the categorized approaches work.

Table 6.1: Tools for scRNA-seq data imputation. Selected methods for data imputation categorized into three main concepts. Distributional assumption is the underlying hypothesis that the data is arisen from. Possible scenarios predicting gene-gene relation are indicated as an output. NB: negative binomial distribution; ZINB: zero-inflated negative binomial distribution.

Tool	Author, Year	Concept	Distributional assumption	Gene-gene relation predictions
knnsmooth	Wagner et al., 2017	smoothing	NB	linear
magic	Dijk et al., 2018	smoothing	low-rank representation	non-linear
saver	Andrews et al., 2018	modeling	NB	linear
dca	Eraslan et al., 2019	deep learning	NB/ ZINB	non-linear

Smoothing-based methods. These imputation methods identify similar cells by their gene expression profiles, for example, by arranging them in a graph structure (i.e. k-nearest neighbor (kNN) graph). The expression values are smoothed or diffused through similar cells. By this, all zero as well as non-zero values are adjusted accordingly.

Model-based methods. This group of imputation methods aims to model zero counts using probabilistic models. Here, the models attempt to differentiate between biological and technical zeros and impute expression values for the latter case.

Deep-learning-based methods. Deep-learning approaches are able to impute non-linear gene-gene relationships. They are able to represent the gene expression matrix in a hidden non-linear latent space. Usually, a normalization step as well as dimensionality reduction (by using the latent space) is integrated within the tools. The imputed values are then estimated from the latent space.

6.2 EVALUATING NETWORK MODELS WITH IMPUTED DATA

Table 5.1 and Table 6.1 provide an overview about selected tools reconstructing gene regulatory networks and data imputation, respectively. The aim is now to evaluate the effect of imputation methods on the predicted networks on different experimental datasets. The combination between imputation and network inference on different datasets results in a cubic evaluation matrix. To manage this we restrict our selection to state-of-the-art computational tools, both for imputation and network inference, that perform most accurately and have been recommended in recent benchmark studies (Hou et al., 2020; Pratapa et al., 2020). Note that we excluded the network reconstruction algorithm SINCERITIES from our analysis as it performs very poorly on experimental datasets. Consequently, we developed a computational pipeline to study seven cell types that were obtained from different single-cell RNA-seq experiments, using four state-of-the-art imputation methods combined with the top performing GRN methods as depicted in Figure 6.1.

Information on the seven cell types was derived from five experimental single-cell RNA-seq datasets: human embryonic stem cell (hESC) (Chu et al., 2016), human hepatocytes (hHep) (Camp et al., 2017), mouse embryonic stem cell (mESC) (Hayashi et al., 2018), mouse dendritic cells (mDC) (Shalek et al., 2014) and mouse hematopoietic stem cells (mHSC) (Nestorowa et al., 2016) that were further separated into the following subtypes: erythrocytes (mHSC-E), granulo monocytes (mHSC-GM) and lymphocytes (mHSC-L). We preselected the datasets according to significantly varying transcription factors and the most highly variable genes across pseudotime.

For the four imputation methods, we chose the following methods summarized in Table 6.1 before: two smoothing-based tools magic (Dijk et al., 2018) and knn-smoothing (Wagner et al., 2017); a Bayesian model-based tool saver (Huang et al., 2018) and a deep-autoencoder based tool dca (Eraslan et al., 2019). We included dca because the authors specifically expect to improve network reconstruction. A baseline model was established using normalized but unimputed data.
As for GRN reconstruction, we selected the following tools: an information-based tool PIDC (Chan et al., 2017), and two tree-based tools GENIE3 (Huynh-Thu et al., 2010) and GRNBoost2 (Moerman et al., 2019). The PPCOR (Kim, 2015) method is based on partial correlations and as such also a contender for a good network reconstruction method. However, PPCOR results on single-cell data are clearly inferior to those obtained with any of the first three methods as shown in Supp. Fig. A2. While we have included PPCOR in this performance comparison, we focus the study of the relationship between imputation and network reconstruction on the other three methods.

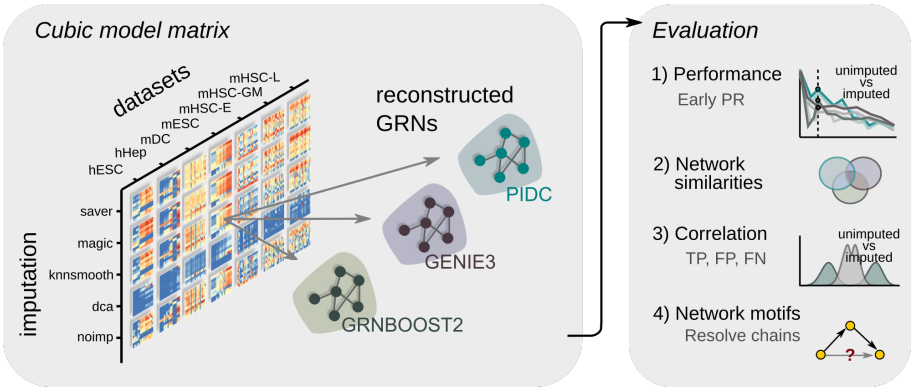


Figure 6.1: Systematic evaluation of network reconstruction from imputed and unimputed data.
Cubic evaluation matrix consists of seven cell types from experimental scRNAseq data, four imputation methods (see text) and three network reconstruction algorithms. Imputed and unimputed ("noimp" in the Figure) scRNAseq data provide input expression matrices which are used by the gene regulatory network (GRN) reconstruction algorithms using the BEELINE framework (Pratapa et al., 2020). We evaluate the performances using the early precision ratios (EPR) and compare network results across different models. Additionally, we inspect the effect of gene-gene correlation on prediction classes (true positives (TP), false positives (FP), false negatives (FN)) before and after imputation, and we search for common motifs within the reconstructed networks. hESC: human embryonic stem cells, hHep: human hepatocytes, mDC: mouse dendritic cells, mESC: mouse embryonic stem cells, mHSC-E, mHSC-GM, mHSC-L: mouse hematopoietic stem cells - erythrocytes, granulo monocytes, lymphocytes.

In the remainder of this chapter we use the term "model" to refer to the combination of a GRN reconstruction algorithm with an imputation method or no imputation, respectively. We obtain the evaluation networks from the STRING database – a functional protein-protein interaction network (Szklarczyk et al., 2019) as well as cell-type specific ChIP-seq derived networks provided by Pratapa et al., 2020. Studying gene regulation we only consider edges outgoing from transcription factors in the reconstructed networks. To evaluate the performance of each network model we use the evaluation framework BEELINE (Pratapa et al., 2020) (see Appendix A.3.4). Furthermore, we inspect the reconstructed network and compare the results with one another.

6.2.1 Imputation does not improve the performance of network reconstruction in general

A compact overview of the results obtained under all the models compared to the STRING network is provided in Figure 6.2. Analyses have been performed on sets of significantly varying transcription factors (TFs) along with 500 and, respectively, 1000 most highly variable genes (HVGs). Each box summarizes results for one GRN reconstruction method. The performance measurements achieved by the respective model on the seven cell types are arranged on a vertical axis. Two performance measures have been computed for each prefiltered gene set correspondingly: the early precision ratios (EPR) (Pratapa et al., 2020) which are shown in the three boxes of Fig. 6.2A,

and the log2-ratios between $EPR_{imputed}$ and $EPR_{unimputed}$ which are shown in the three boxes of Fig. 6.2B. EPR refers to the number of true positive interactions within the top-k network normalized by the network density. Here, k refers to the number of positive interactions found in the evaluation network. An EPR of 1 indicates a random predictor. The second performance measure compares the performance of an imputation method relative to the performance of not using imputation. Here, a value of zero means no change, while a negative value indicates a detrimental effect of the imputation.

The EPR scores for unimputed data that were reported by Pratapa et al., 2020 could be reproduced with minor deviations in our analysis. The EPR scores are illustrated as a dashed line in Figure 6.2A. Results vary strongly with the datasets; the scores range from approximately 2 (for the mDC dataset) to 8 (for mHSC-GM), with less variation across GRN reconstruction algorithms. Applying imputation with either dca, knnsmooth or magic, does not improve the performance in any of the GRN reconstruction methods. While in mDC data the performance scores in each model scatter around the unimputed model, in mHSC-GM data the performance scores vary strongly, dropping from 8 to just below 5 for the magic/GENIE3 model. As pointed out above, for PPCOR we observe considerably lower performance scores compared to the remaining GRN algorithms (Supp. Fig. A2). The respective EPR scores indicate predictions comparable to a random model that we decide to exclude PPCOR from further evaluations.

Focusing on the change of performance due to imputation as measured using the log2-ratios between imputed and unimputed EPR scores, we observe that only saver is able to improve the performance (Fig. 6.2B). The saver/PIDC model achieves log-fold-ratios up to +0.5 in 5 out of 7 datasets and 2 out of 7 datasets combined with GENIE3 or GRNBoost2. All other imputation methods worsen the performance with log-fold-ratios down to -1 which represents a performance decline of 50% in comparison to the unimputed model. Generally, the performance results regarding the number of most highly variable genes are highly consistent suggesting that the predictions are irrespective of the number of genes selected as an input.

Furthermore, we use cell type-specific networks derived from ChIP-seq data as an evaluation network (Supp. Fig. A2). Here, absolute EPR scores report very poor performances close to or worse than a random predictor regardless of the model or the number of input genes across all datasets. Thus, the ChIP-seq network does not serve us well for distinguishing between methods in terms of their accuracy. The STRING database, on the other hand, may contain indirect interactions reported in the protein-protein interaction data. We will return to this issue below in the context of network motif analysis. Nevertheless, independent of the evaluation network we do not see an improvement of GRN reconstruction if imputation has been used in advance.

We further asked the question whether data quality as given by sequencing depth is a determinant of the success of imputation prior to GRN reconstruction. To answer this, we simulated cells *in silico* by downsampling the gene counts of the given experiments to 60% of their sequencing depth, thereby lowering the detection rate (Supp. Fig. A3). The hope would be that imputation has a more beneficial effect in these simulated data sets as compared to the original, higher quality data. However, similar results as above were obtained when we subjected the lower quality *in silico* data to our analysis pipeline (Supp. Fig. A4). Like with the original datasets, saver/PIDC

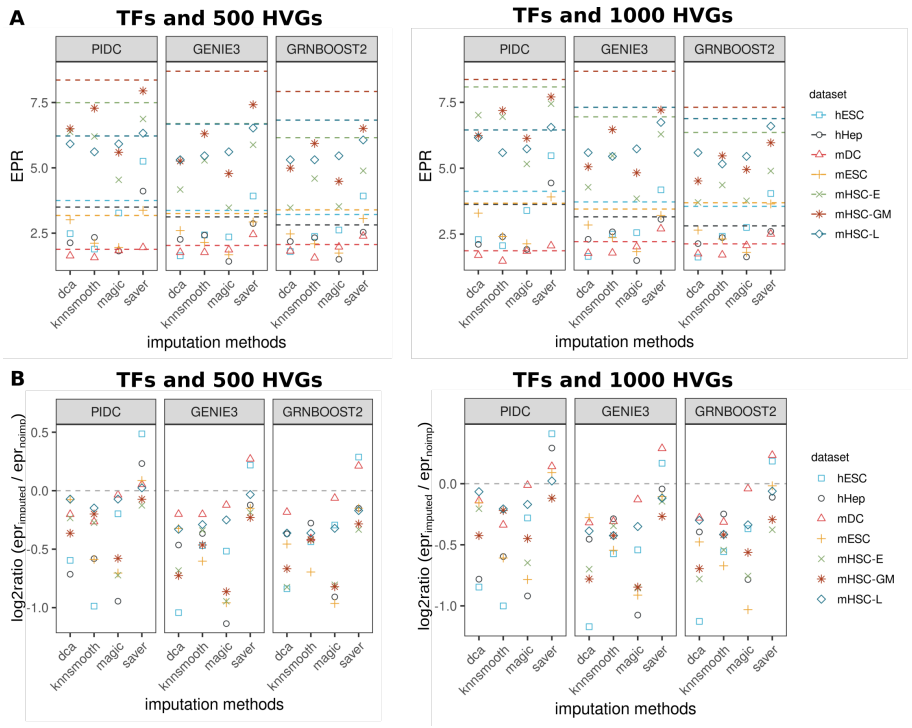


Figure 6.2: Impact of imputation on network reconstruction performances. Results under all models with two different sets of genes compared to the STRING network. (A) Absolute EPR scores across imputation methods (x axis label) and GRN inference algorithms (box) on seven different cell types (coded by shape and color). Dashed lines represent EPR scores obtained without imputation. EPR = 1 corresponds to a random predictor. (B) log2-ratios between EPR scores obtained using imputed and unimputed data. Log2-ratio = 0 represents no change in performance (grey dashed line) after imputation.

obtain the highest improvements compared to the downsampled unimputed datasets. Nonetheless on downsampled data, dca, knnsmooth and magic are able to improve performance in some of the tested datasets, although not consistently.

Overall, our results demonstrate that model performances are highly dataset-dependent. Applying imputation on the original data resulted mostly in a drop of performance of GRN reconstruction compared to the unimputed model, although potentially improving performance on low-quality data tested *in silico*.

6.2.2 Imputation method rather than GRN method determines results

The analysis presented in the preceding section raises the question how strongly either the choice of imputation method or of network reconstruction algorithm affects the results. To answer this question we first address the variability in results when varying either the one or the other, and

then study similarity among computed networks across the models.

With regard to the performance variability, we compare the variance of EPR log-fold-ratios under a fixed GRN reconstruction algorithm while varying across imputation methods, and, vice versa, varying the GRN algorithm while keeping the imputation method fixed. As Figure 6.3A shows, EPR log-fold-ratios vary much more strongly across the imputation methods than across GRN methods (wilcoxon-test p-value ~7.86×10-6). Since this analysis aggregates all datasets jointly, it discards the differences between datasets. Comparing e.g. hESC and mHSC-L, we see large differences between the distributions of variances across imputation methods and GRN algorithms, respectively. To resolve this, we perform an analysis of variance (ANOVA) with respect to the EPR log-fold-ratios for each dataset separately. The results give evidence that imputation has a larger contribution to the variance of performance scores compared to GRN algorithms, prevalent in all datasets except mHSC-L (Supp. Tab. A1). This implies that the choice of imputation method determines the quality of results to a larger degree than the choice of GRN reconstruction algorithm.
A direct consequence of this observation is the suspicion that the topology of the predicted networks may also be largely determined by the imputation method and to a lesser degree by the GRN reconstruction method. To test this, we inspect the overlap among the 500 most important gene-gene interactions of the computed networks. Here, we calculate pairwise similarity scores using the Jaccard index and use it to hierarchically cluster the networks. We found that networks tend to cluster with respect to imputation methods but not GRN methods (Fig. 6.3B). To make this more precise, we use as a measure of cluster purity the adjusted rand index (ARI) (Gates et al., 2017; Hubert et al., 1985). ARI coefficients calculated across the seven different cell types show higher cluster purity when labeled with imputation methods as opposed to network reconstruction algorithms (Fig. 6.3C).

We conclude that the imputation method largely determines model performance, leaving little influence to the subsequent GRN reconstruction algorithm. The choice of imputation method further biases the outcoming network leading to little consensus across the most important recovered gene-gene interactions as computed based on different imputation methods.

6.2.3 Inflation of gene-gene correlations and its impact on the network topology

Based on the reported results, we examine how imputation generally affects gene-gene correlation coefficients. Although not all network reconstruction algorithms use correlation-based measures to recover interactions, we still use Pearson's correlation coefficient as a proxy for the association between two genes. Subsequently, we will investigate whether the interactions within the reconstructed networks affect the global network structure.
Exploring the overall distributions of gene-gene correlations after imputation on single-cell RNA-seq data we observe a strong enhancement in gene-gene correlations (Fig. 6.4A). Generally, gene-gene correlations go from almost no correlation when computed using unimputed data to very good anti- and positive correlations due to imputation. Here, magic leads to the most extreme enhancement. Surprisingly, even the unimputed distribution within the mDC data is skewed

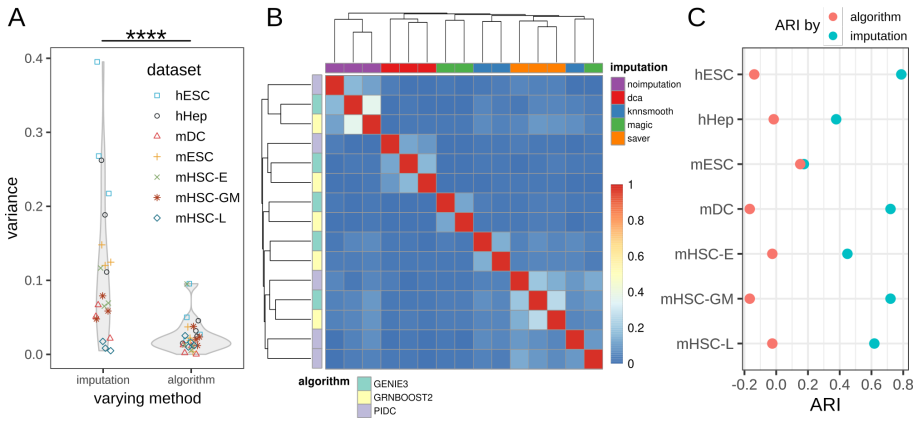


Figure 6.3: Variability in network results largely stems from imputation methods.
(A) Variance distribution of EPR scores across imputation methods. Left violin plot keeps the GRN algorithm fixed and depicts the variances in EPR log-fold-ratios for each dataset across the imputation methods. Right violin plot shows the variances for fixed imputation methods. **** corresponds to p-value ≤ 0.0001 by wilcoxon rank sum test. (B) Clustered heatmap of network similarities measured by Jaccard index within top 500 reported interactions. Columns (horizontal axis, above) are color-coded by imputation methods. Rows (vertical axis, left) are color-coded by network inference algorithms. More pure clusters are obtained by imputation than by GRN algorithm. (C) Adjusted rand index (ARI) obtained for clustering results in each cell type by annotation label "algorithm" (pink) and "imputation" (blue), respectively.

towards positively correlated values. We have checked that this is not due to selection of the most highly variable genes, but rather already present in the dataset. Figure 6.4B exemplifies the association between three genes before and after imputation, transforming very weak correlations to almost perfect (anti-)correlations. This particular set of gene interactions was observed in the top-k network computed with GRNBoost2 on hESC data, comparing no imputation with dca imputation. Indeed, we commonly find such associations across different datasets and imputation methods.

In order to see what impact this enhancement of correlation has on the network structure we next investigated the network density after imputation in relation to the unimputed data using log-ratios (Fig. 6.5A). Here, we looked at the top-k networks according to the EPR score. Imputation methods alter the network densities with log-ratios ranging from -0.5 and +0.5 in hESC, hHep, mDC and mESC data, except for saver and PIDC in hESC data with a slightly higher value of 0.59. For the three subtypes of mHSC data we observe larger changes in network density reaching log-ratios beyond ±1. Especially here, imputations combined with GENIE3 and GRNBoost2 lead to a sparser network whereas all combinations of imputation methods with PIDC lead to a denser network structure. We assume that this is due to redistribution of edges occurring in the tree-based algorithms which is also reflected in the node degree distribution (Fig. 6.5B).
Before imputation we observe a heavy tail node degree distribution predominantly in GENIE3 and GRNBoost2 indicating the presence of many hub nodes. After imputation the heavy tail disappears when using dca, magic and knnsmooth while it still exists when using saver. Generally,

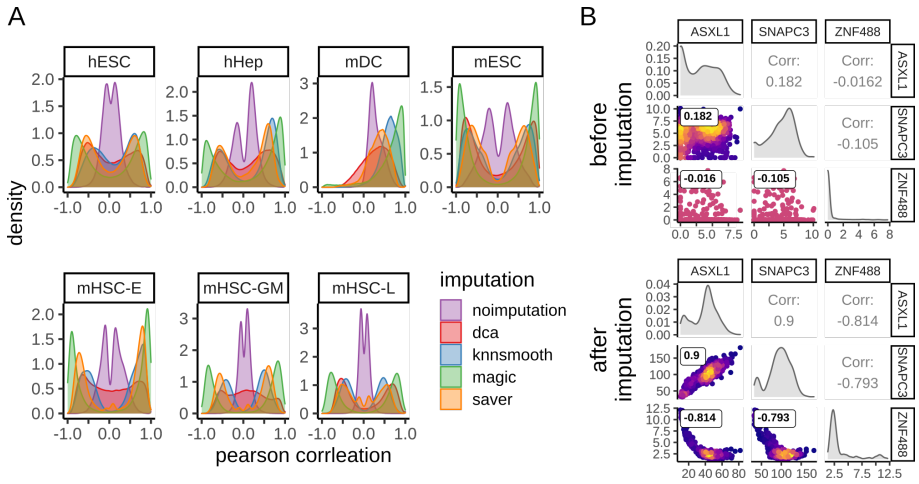


Figure 6.4: Gene-gene correlations before and after imputation.
(A) Gene-gene correlation distributions obtained in each cell type color-coded by imputation method among top 500 most variable genes and significantly varying TFs. (B) Paired density scatter plots before and after imputation with dca. GRNBoost2 reported the pairwise interactions between ASXL1, SNAPC3 and ZNF488 among the top-k network after imputation in hESC data.

PIDC does not lead to this structural change in node degree distribution.

As a conclusion, the enhancement of gene-gene correlations due to imputation appears to lead to notable changes in the topology of the predicted gene networks.

6.2.4 Increased correlation values lead to inflation of false positive predicted interactions

Since we have observed that imputation may decrease the performance of GRN network reconstruction, we attempt to understand how the altered correlations in imputed data affect network reconstruction. To this end, we explore the change of edge ranks and correlation values of the reported (i.e., positively predicted) and missed (i.e., negatively predicted) interactions.
Overall, the ranks of true positive (TP) interactions reported in the unimputed data change significantly after imputation (Fig. 6.6A, Supp. Tab. A2, Supp. Fig. A6). Some of the previously reported TP interactions could be recovered after imputation. Nevertheless, the majority of previously reported TP interactions shift after imputation towards the end of the gene interaction ranking list, and are considered less important. As a consequence, other interactions become more important. Therefore, we look at the change of correlation of positively predicted interactions before and after imputation. Figure 6.6B (and Supp. Fig. A7) show scatter plots of gene-gene interactions with the absolute values of correlation coefficients before imputation on the horizontal axis and the correlation coefficient after imputation on the vertical axis. For each model, red dots are the true positive interactions, yellow are the false positives, and blue are the false negatives. The general shape of the scatter plot reiterates the observation that correlation coefficients tend to get enhanced by imputation. For each class we computed regression lines. For better recognition of true positives

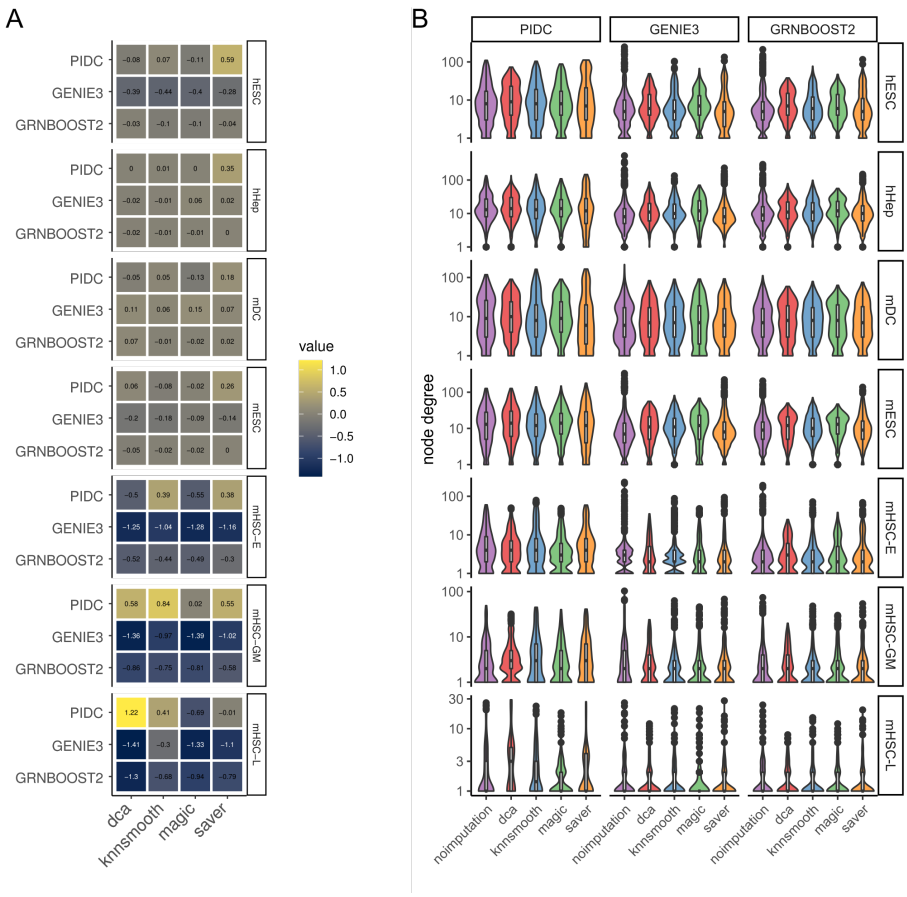


Figure 6.5: Structural changes in inferred networks.
(A) Change of node density before and after imputation. Log2 ratios between density (after imputation) and density (before imputation) are color-coded. Positive values represent a denser whereas negative values represent a more sparse network with respect to the unimputed model. (B) Node degree distribution across all models. Y-axis is log-scaled.

after imputation, one would hope for the TP regression line (shown in red in Fig. 6.6B) to lie well above the others – which is not really the case. We generally observe a strong enhancement of correlations as indicated by the height of the intercept of the regression lines. In 11 out of 12 cases the regression lines for both true and false positive predictions are almost congruent with each other. Note that the red color dominates the other ones and the dots below a red one are not visible.

Interestingly, we see remarkably different regression lines if we take the false negative (FN, blue) interactions into account. The majority of FN correlations remain low after imputation, as indicated by the height of the intercept in Fig. 6.6B. Presumably, the FN correlation values that actually get enhanced get lost in the background due to the inflation of FP correlations in the inferred top-k network. Thus, the boost of correlation values makes it harder for GRN reconstruction methods to separate the actual signal from the background.

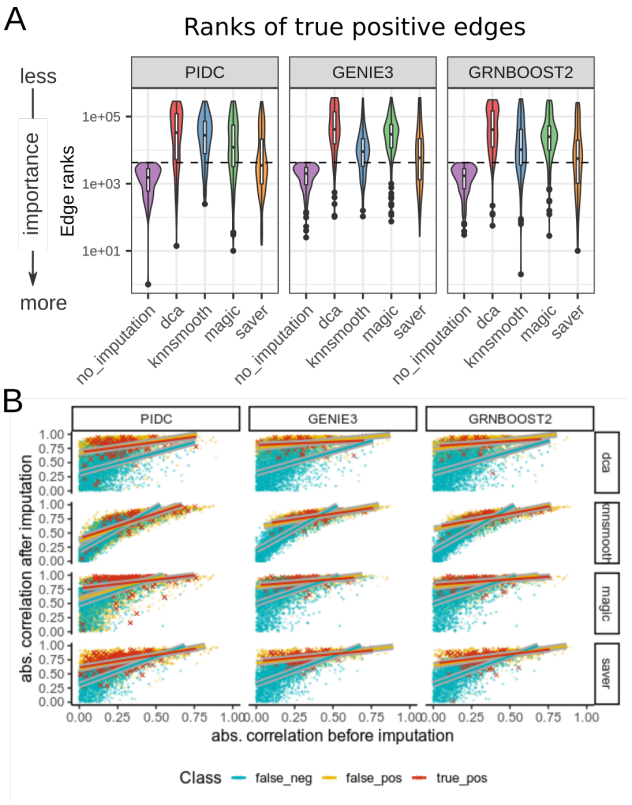


Figure 6.6: Impact on predicted interaction classes after imputation.
(A) Change of edge ranks in true positive (TP) interactions identified by unimputed model after imputation in hESC data. Dashed line indicates the rank threshold corresponding to the top-k network. Interactions below the dashed line represent TP within the respective model. Low edge ranks represent highly important interactions. (B) Scatter plots comparing correlation values between genes before vs after imputation. Each scatter plot corresponds to one model in hESC data. True positives are red crosses, false positives are yellow dots, and false negatives are blue dots. For each scatter plot we fit one regression line for TPs, FPs, FNs, respectively, with the corresponding color. For visualization purposes we added a background color to the lines to better distinguish the line and the dots. Positively predicted interactions differ clearly from FN interactions.

Many GRN reconstruction methods have the goal of distinguishing direct interactions from transitively inferred ones (Ghanbari et al., 2019). Therefore, we tested whether the GRN reconstruction algorithms analyzed in this study are able to make the necessary distinction. Given three genes X, Y, and Z where X is correlated with Y, and Y is correlated with Z, these genes form a network chain. However, oftentimes by transitivity these associations seem to imply a correlation between X and Z, thus forming a network loop. Generally, in network theory it is challenging to distinguish chains from loops and algorithms deal differently with it. PIDC constrains the inferred interactions based on a confidence score to discriminate between direct and indirect interactions. GENIE3 and GRNBoost2 allow the user to set a parameter for filtering out presumably indirect interactions. In this context, we analyze how the models deal with the identification of network chains from imputed data among the top-k networks. Errors are counted if a supposedly false loop is detected (false negative, FN) or a chain is detected instead of a loop (false positive, FP). However, STRING is less suited as an evaluation network in this context because false positive counts might be overestimated upon comparison to STRING. This is due to the fact that the STRING database is not designed to contain only direct interactions. For example, protein complexes are reported in STRING and may contain indirect associations. Therefore, in this analysis we use the ChIP-seq derived networks as the more appropriate evaluation networks. Figure 6.7A shows true positive (TP) counts together with the error counts in hESC data. Here, we observe mainly lower motif counts after imputation. In general, low count numbers in the motif search are indicative for isolated edges between a gene pair. Hence, the algorithms detect fewer connected edges among the top-k networks.
In order to measure the performance between true and false predictions we also calculate the true

positive rates (TPR) and false discovery rates (FDR) for each network inference and imputation method applied to each dataset (Fig. 6.7B). Ideally, the values for TPR should be higher (yellow) while the values for FDR stay low (purple). Comparing the TPR and FDR scores after imputation, however, we do not see systematic differences. We conclude from this observation that the performance of network motif detection among the top-k networks does not seem to be affected by imputation. Hence, either imputation methods do not necessarily induce transitive correlations or the network reconstruction methods deal well with transitively induced correlations.

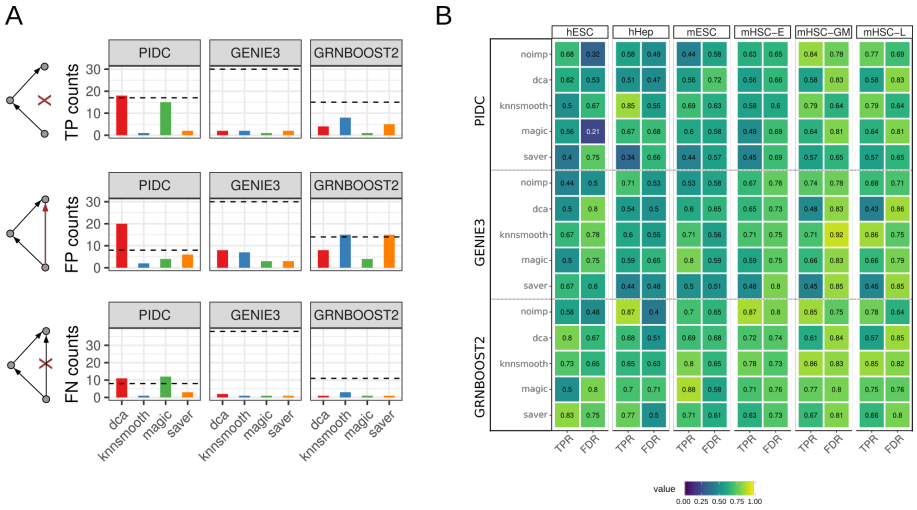


Figure 6.7: Network motif counts across models.

(A) Counts of positively and negatively predicted network chain motifs in hESC data for each model. TP network chains agree both in prediction and evaluation networks (ChIP-seq derived network). FP network chains are falsely positively predicted chains being actual feed-forward loops in the evaluation network. FN network chains are falsely predicted as being feed-forward loops when they are actually network chains in the evaluation network. (B) TPR and FDR scores for network chain motifs obtained by statistics in A). mDC dataset is not included as no motifs could be found among the top-k network. Ideally, TPR values should be close to 1 whereas FDR values should be close to 0.

6.3 DISCUSSION

The advent of single-cell transcriptomics has rekindled the interest in reconstructing gene regulatory networks from transcriptomics data, primarily for two reasons. Firstly, it is of great interest to study regulation from single-cell data in the hope to eventually uncover how, e.g., differentiation processes proceed. Secondly, the main obstacle in gene network reconstruction from bulk transcriptome data appears to be the low number of available samples in comparison to the large numbers of genes. For example, simulations have demonstrated that high quality reconstruction of gene networks requires a much larger number of samples than the number of genes (Ghanbari et al., 2019). Seeing each single cell as a sample, the expectation arose that single-cell transcriptomics would solve this conundrum by providing a sufficiently large number

of samples, thus putting high quality network reconstruction within reach.

It was sobering for us to see that due to the sparse nature of single-cell RNA-seq data, individual cells cannot contribute as much information to network reconstruction as bulk samples. Indeed, preprocessing of single-cell data for data analysis is crucial (Vieth et al., 2019), and is implemented in many computational pipelines. Imputation has become a possible element of this preprocessing in the hope it would supplement the missing information. In this study we have, however, demonstrated that the choice of imputation prior to GRN reconstruction influences the results in a two-fold manner: First, it affects the performance of network reconstruction leading to highly variable accuracies and, secondly, the reconstructed network is determined more by the imputation method than by the choice of network reconstruction method.

The focus of our work on the interplay of the imputation step with GRN reconstruction clearly also limits the scope of our work: We have not attempted to compare GRN methods as such, nor to improve GRN reconstruction. Many other publications are dedicated to these issues, with GRN reconstruction being a particularly hard problem as evidenced by the overall meager results that can be obtained (Chen et al., 2018a; Pratapa et al., 2020). Still, what has been clearly understudied is the interdependence between imputation, which is routinely done in single-cell data analysis, and GRN reconstruction.
We have systematically evaluated the effect of imputation on GRN reconstruction using experimental scRNAseq data on seven cell types. In agreement with previous studies, we see that imputation may boost gene-gene correlations in a questionable way, thereby introducing false positive edges in a network (Breda et al., 2019; Steinheuer et al., 2021). Steinheuer et al., 2021 evaluated the impact of data imputation on network inference via a gene correlation analysis using simulated data. There, the authors downsampled bulk RNA-seq data, applied imputation methods and compared the gene module preservation and edge recovery upon imputation. Similar to our observation they notice a higher number of false positive interactions after imputation.

We have provided evidence that these false positives may lead network algorithms to reinforce dependencies that have been introduced by imputation. For example, regression-based methods like GENIE3 and GRNBoost2 will be strongly predisposed to including imputation-generated correlations into a network. Table 6.1 recalls which assumptions imputation methods make with respect to signal distribution and the linear or non-linear nature of interactions. Likewise, GRN reconstruction algorithms are each based on their own respective assumptions (Table 5.1). This may lead to reinforcement of imputation decisions or, generally, to the identification of wrong gene-gene dependencies. Andrews et al., 2018 have warned of this circularity before, albeit in the context of differential expression analysis (Andrews et al., 2018). Consistent with our findings, Andrews et al., 2018 showed that saver introduces the smallest number of spurious gene-gene correlations. We speculate that the combination of saver/PIDC works well because saver is a model-based imputation method and PIDC is a mutual-information based algorithm discretizing the data beforehand; the two approaches follow independent assumptions complementing one another, thus avoiding the use of redundant information.

In this study we have tested our hypothesis on experimental datasets with fairly large library sizes and gene detection rates (Supp. Fig. A3). In order to test our hypothesis on more shallowly

sequenced single-cell experiments we *in silico* lowered the detection rate introducing more zero counts. These results again show that using saver with PIDC improves results in most cases. Thus, if single-cell data is too sparse to avoid imputation altogether, we recommend the use of saver and PIDC. It should be noted, though, that we are not discouraging imputation in general. There may be many other applications that are not studied here, where imputation can be useful, depending on the type of analysis that is subsequently performed.

We believe that the described interdependence among processing steps within a data analysis pipeline is exemplary for many data analysis tasks. Software is generally being built to allow the user to freely combine algorithms, each dedicated to a particular step of the analysis. Little attention is given to the possible influences one algorithm might have on the behavior of the other. We are not referring to a syntactic interaction in terms of data structures or variables passed, since good, modular software design will exclude such conflicts between processing steps. Much rather, as we demonstrated for imputation and GRN reconstruction, decisions taken within one algorithm may predispose the results that can be obtained in a downstream analysis step. Thus, user friendliness in pipeline design allowing the free combination of algorithms may carry substantial risk with respect to the scientific validity of data analysis results.

Limitations and future insights

The findings of the study presented have some limitations that we want to address here. Above all, the wiring of the cell is still not fully understood and thus the choice of gold standard dataset for GRN reconstruction will necessarily remain problematic. We follow the literature in that we use STRING, containing protein-protein interactions, and cell-type specific datasets of ChIP-seq derived interactions. However, all methods studied have difficulties in identifying interactions from the latter dataset. Thus, we learn more from the STRING database although it does not contain cell type specific information.

There is further room for improvement in exploiting pseudotime derived from single cell data. However, the methods geared towards this goal follow different principles and Pratapa et al., 2020 have shown that network reconstruction algorithms using pseudo time information are very sensitive to the temporal ordering of the cells. Thus, in addition to studying the dependence between imputation and GRN reconstruction, it would also be necessary to study the interplay between pseudotime reconstruction method and GRN reconstruction. The preeminent question following from our study is clearly how one can best utilize the large number of cellular transcriptomes for the purpose of GRN reconstruction without initially relying on imputation.

7 CONCLUDING REMARKS

In this thesis we introduce how transcriptome profiling evolve from experiments based on bulk technologies to single-cell RNA-seq technologies (Chapter 2) and describe what computationally challenges and steps need to be considered in analyzing single-cell RNA-seq data (Chapter 3). We show how we can assess the variability within biological experiments and in a simulation framework (Chapter 4) and how we study gene regulation in the context of gene regulatory networks as well as how preprocessing may influence the network prediction performance (Chapter 5-6).

The research field of single-cell transcriptome profiling emerges rapidly allowing for sequencing up to millions of cells. With the increasing amount of profiled sample sizes using single-cell technologies, the expectation of resolving gene regulatory networks appear. However, successful reconstruction of gene regulatory networks depend on multiple factors: (i) sufficiently large biological variability across samples and (ii) a good data quality ensuring to have sufficient information about gene expression level, i.e. the rate of detected genes in a single-cell experiment.

In Chapter 4 we show how challenging it has been to reveal the biological heterogeneity that we are interested in a single-cell experiment. This study has shown how the technical variability covered the biological signal preventing us from studying differentially expressed genes across Influenza A infected single cells. We were able to identify defective interfering particles that co-infect cells and get amplified as part of the viral replication procedure. The co-infection and amplification of the defective interfering particles might compete with viral transcripts leading to a decrease in the viral reproduction rate. Although our results have not been able to reveal any explanation arising from the host cells, we want to address and emphasize on the difficulties in analyzing the single-cell experimental data. These difficulties have led us to implement a simulation framework to study the variability present in single-cell transcriptome data. We show that in our simulated dataset the technical variability, in the course of the sampling process, arises from the Poisson distribution. The simulation framework has allowed us to study other technical characteristics as the integration of bulk and single-cells into the same topology.

In fact, the methodology of projecting bulk and single-cell samples in the same topological embedding using UMAP has been used in other research studies within our group as well. Virginie Stanislas has applied this method on one of her research studies projecting immune cell (T-cell) bulk RNA-seq samples onto a pre-calculated embedding using single-cell RNA-seq data of peripheral blood mononuclear cells (PBMCs) that has been filtered for T-cell population. In another project Aybuge Altay has used this procedure to project bulk and single-cell ATAC-seq data obtained from PBMC samples onto the same embedding. ATAC-seq is a technology that allows the quantification of accessible DNA regions referring to open chromatin regions. Thus, the methodology of projecting bulk and single-cell samples is not only applicable as part of our simulation study but can be also transferred to experimental datasets, for both RNA-seq and

ATAC-seq data.

Furthermore, we have investigated similarity measures and have observed a bias towards samples in which the gene detection rate was high. Cells with a higher amount of detected genes have a "better" similarity score when we compare them with its reference bulk sample. This might pose a risk because of this purely technical factor samples with a generally higher detection rate might appear more similar than those with a lower gene detection rate. Hence, this needs to be taking into account when measuring cell similarity measure on read count data only. Overall, this chapter provides a solid basis to understand and assess the heterogeneity regarding the technical variability. Along this investigation other important technical features could be studied that needs to be considered when analyzing single-cell transcriptome data.
The next aim of the thesis has been to study gene regulatory networks using single-cell transcriptome data. We have reconstructed gene regulatory networks using the *neighborhood selection* procedure. We have developed a methodology based on a two-fold selection criteria to select for the hyperparameter λ. We have successfully applied this method on several scenarios simulating dynamical processes evolving through a continuous time-dependent trajectory. This provides a fundamental analysis for future studies to reconstruct gene regulatory networks that evolve dynamically capturing different stages of cell differentiation. Our attempts to apply the neighborhood selection procedure on experimental data has not shown encouraging results. We reason this moderate performance due to the high numbers of zero thus the increased level of technical noise. For this reason, we have applied imputation but observed a drop in the prediction performance relating to a random predictor. This observation has guided us to the question if imputation generally improves or hinder the reconstruction of gene regulatory networks. Due to the lack of consensus pipeline and conflicting attitudes whether or not imputation facilitates network reconstruction, we systematically evaluated the effect of data imputation on the performance of network inference using state-of-the-art algorithms. The insights gained from the studies are valuable and let us conclude that adding information by imputation might bias and obscure network structures towards the imputation method that has been applied prior to the network algorithm.

Hence, for future studies we recommend avoiding the use of imputation but to rather aggregate closely related cells to form small-sized populations of cells (so called pseudo-bulks). As a result, the pseudo-bulk samples have eventually a smaller fraction of zero counts such that network reconstruction algorithms do not fail in performing due to the large amount of zero counts. However, it is important to keep in mind that the biological variability needs to be maintained as much as possible while "filling up" the dropouts when aggregating the cells. In an extreme scenario of repetitively aggregating samples, one indeed ends up with highly informative samples with almost no zero counts but only little variation across the small sample sizes, similarly to bulk RNA-seq experiments. Thus, during the aggregation process one needs to find a trade-off between information/signal content versus biological variability in order to be able to reconstruct gene regulatory networks successfully.

In conclusion, we have dissected single-cell transcriptome data in terms of heterogeneity and used the biological heterogeneity in order study gene regulatory networks. This thesis contributed

valuable insights for the single-cell community and provides a fundamental basis to develop further algorithms reconstructing gene regulatory networks using single-cell transcriptome data.

ABBREVIATIONS

AUROC area under receiver operating characteristic curve

bp base pair

cDNA complementary DNA

CLP common lymphoid progenitor

CMP common myeloid progenitor

CPM counts per million reads

CV cross validation

DIP defective interfering particle

dNTP deoxynucleotide

ddNTP dideoxynucleotide

EP early precision

EPR early precision ratio

ERCC External RNA Control Consortium

FACS fluorescence-activated cell sorting

GMP granulocyte/ macrophage progenitor

GTEx Genotype-Tissue Expression

GRN gene regulatory network

HSC hematopoietic stem cell

HVG highly variable gene

IAV Influenza A virus

KL Kullback-Leibler

kNN k-nearest neighbor

MI mutual information

MEP megakaryocyte-erythrocyte progenitor

mHSC mouse hematopoietic stem cell

MOI multiplicity of infection

MPP multipotent progenitor

ODE ordinary differential equation

PCA principal component analysis

PCR polymerase chain reaction

PID partial information decomposition

PFU plaque forming unit

STRT single-cell tagged reverse transcription

sci-RNA-seq single-cell combinatorial-indexing RNA-seq

TF transcription factor

TPM transcripts per million reads

t-SNE t-distributed stochastic neighbor embedding

UMAP Uniform Approximation and Projection

UMI unique molecular identifier

LIST OF FIGURES

LIST OF TABLES

A APPENDIX

A.1 EXPERIMENTAL PROCEDURES AND DATA PROCESSING

Procedure for single-cell RNAseq in IAV-infected cells. The single-cell lysates (5 µL) were transferred to a 96-well plate, and subjected to a protocol for Smart-seq2 that allows for the generation of full-length cDNA and sequencing libraries, according to Picelli et al., 2014. Briefly, 0.1 µL of 1:80,000 External RNA Control Consortium (ERCC) RNA spike in controls, 1 µL of dNTPs (10 mM), and 0.1 µL of oligo-dT30VN (5'-AAGCAGTGGTATCAACGCAGAGTACT30VN-3'; 100 µM) were added to the 5 µL of cell lysate (on ice). The mixture was incubated for 3 min at 72 °C followed by 10 min at 10 °C. After hybridization of oligo-dT to the polyA tail, reverse transcription was performed: to each well a master mix containing 0.25 µL RNase-Inhibitor (40 U/µL; final amount/rxn 10 U), 3.4 µL SuperScript First Strand Buffer (5x; final amount/rxn 1x), 0.85 µL DTT (100 mM; final amount/rxn 5 mM), 3.4 µL Betaine (5 M, final amount/rxn 1 M), and 2.04 µL MgCl2 (50 mM; final amount/rxn 6 mM) were added. To start the reverse transcription, at the very last moment, 0.2 µL template switching oligo (TSO: 5'-AAGCAGTGGTATCAACGCAGAGTACATrGrG+G-3') (100 µM; final amount/rxn 1 µM) and 0.7 µL of SuperScript II reverse transcriptase (200 U/µl; final amount/rxn 140 U) were added and the following incubation program was started in a thermocycler with a heated lid: 90 min of incubation at 42 °C, 10 cycles of 50 °C for 2 min and 42 °C for 2 min, and 15 min of incubation at 70 °C for enzyme inactivation.
Reverse transcription was followed by a preamplification step that was performed on magnetic Agencourt Ampure XP beads (Thermo Scientific): 17 µL of beads was mixed with the RT mix and incubated for 8 min. The plate was put on a magnetic stand for 2 min and the supernatant discarded. To each well, 16 µL of PCR mastermix was added (8 µL of 2x KAPA Hifi Mix, 0.2 µL of ISPCR-primer (5'-AAGCAGTGGTATCAACGCAGAGT-3', 10 µM) and 8 µL of nuclease free water) and the following PCR program was run: 98 °C for 3 min, 18 cycles of 98 °C for 20 s, 67 °C for 15 s, and 72 °C for 6 min, followed by a final incubation at 72 °C for 5 min.
For clean-up, 16 µL of Ampure XP beads was added and incubated for 8 min. After placement on a magnetic stand, the supernatant was discarded and the beads washed twice with 200 µL of freshly prepared ethanol. Beads were resuspended in 10 µL EB and after further incubation on the magnetic stand, the supernatant containing the DNA was used for library preparation following Illumina's Nextera XT protocol.
Therefore, we used 1/5 of the recommended volumes: 2 µL of Tagment DNA (TD) buffer (2x), 1 µL of Amplicon Tagment mix were mixed with 1 µL of the cDNA and incubated for 4 min at 55 °C. Then, 1 µL of NT buffer was added and incubated for 5 min at room temperature. Adapter ligated fragments were barcoded and amplified by adding 3 µL Nextera PCR master mix, and 1 µL of each index 1 and index 2 primers by applying the following cycling protocol: 72 °C for 3 min, 95 °C for 30 s, 15 cycles of 95 °C for 10 s, 55 °C for 30 s, 72 °C for 30 s, and a final incubation of 72 °C

for 5 min. Barcoded libraries were pooled and cleaned up using 0.6 volumes of AmpureXP beads. Beads were washed twice with 80% ethanol, eluted in 300 µL EB and a further cleanup was performed by adding an additional 180 µL of beads followed by two washes with 80% ethanol. Beads were resuspended in 100 µL EB. Quality controls were performed involving Qubit quantification (Thermo Fisher), Bioanalyzer size assessment (Agilent), and qPCR (Roche: KAPA library quantification kit). Sequencing was performed on a full lane of the Illumina HiSeq2500 system in PE50 mode.

scRNAseq data processing and quality control. Gene expression was quantified by Salmon (Patro et al., 2017), version 0.7.2 including the parameter `libType = IU`, `–posBias` and `–gcBias`. The transcriptome index was built using the Ensembl version 86 Canis familiaris (genome assembly CanFam3.1) cDNA sequences, the genome of IAV of strain PR8, and the sequences of the ERCC RNA spike-ins. For the coverage analysis, STAR (version 2.5.2a) (Dobin et al., 2013) was used in the paired-end and single-end mode, allowing a minimum chimeric segment length of 10 (chimSegmentMin = 10). Other parameters used for STAR:

```
--outFilterMultimapNmax 5
--outFilterScoreMinOverLread 0.25
--outFilterMatchNminOverLread 0.25
--outSJfilterOverhangMin 10 10 10 10
--outSJfilterCountUniqueMin 1 1 1 1
--outSJfilterCountTotalMin 1 1 1 1
```

As a measure of quality control, a sequencing depth of more than 150,000 reads and an ERCC spike-in accuracy of 0.75 was considered. The accuracy was calculated by the Pearson's correlation coefficient between the known concentration and the measured expression level. Additionally, samples with at least 10,000 reads mapping to PR8 in the deletion junction analysis were considered.
Salmon quantifies expression level by transcripts per millions (TPM), which includes the ERCC spike-ins. By removing the ERCC spike-ins and scaling the expression values to a million mapped reads, we obtained the expression level from the endogenous transcripts. Genes were filtered out, which were detected (TPM ≥ 1) in less than five samples.

Analysis of Deletion Junctions. Absolute insert sizes of mate pairs mapping to PR8 were extracted from bam files. We calculated the log2 ratios between the number of large insert sizes (>1000 bp) and small insert sizes (≤1000 bp) on PR8 S1, S2, and S3 with a pseudocount of 1x10-7, avoiding the logarithm of zero. In order to identify the deletion junctions by their exact position, sequence alignment information of split or chimeric reads spanning the junction were used. To obtain the chimeric read information, we first ran STAR using the single-end mode for each read pair separately reducing the alignment artifacts. Next, the two Chimeric.out.junction output files were joined and chimeric reads spanning the junction were counted. Finally, we calculated the deletion junction distance considering the ambiguous split positions by merging chimeric reads with the same distance spanning the same locus with +/- 3 bp difference. Regarding the viral bulk population, junctions were considered that had a distance >1000 bp and covered by >10 reads. For IAV-infected single cell experiments, we included junctions fulfilling the above condition or if junctions were detected in the viral bulk population. Read counts were normalized by counts per millions (CPM).

Data and Software Availability. Collection of next-generation sequencing (NGS) data related to this publication is under BioProject PRJNA590388. The link for the repository that includes the computational analysis is `https://github.com/lylamha/influenza_sc`.

GTEx data collection and processing. GTEx gene expression data was downloaded from the GTEx portal `https://gtexportal.org/home/datasets` on the 09/01/2019. The respective filename was `GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_reads.gct`. The matrix consists of 11,690 samples and 56,202 genes.
First, we normalized the samples using z-score (mean=0 and standard deviation=1) and applied principal component analysis (PCA). Secondly, we applied UMAP using the first 50 principal components and lastly visualized the data in the 2-dimensional UMAP embedding.

We identified ambiguous samples by using a clustering approach. Using the shared nearest neighbor algorithm (by `'dbscan'` R package version 1.1.3) we clustered the samples using the 2-dimensional UMAP embedding and identified the tissue dominating each cluster with a frequency of 90%. Samples falling into that cluster but are different from the dominating tissue were discarded from further analysis. This resulted in a matrix with 7,219 samples.

A.2 NETWORK RECONSTRUCTION WITH NEIGHBORHOOD SELECTION

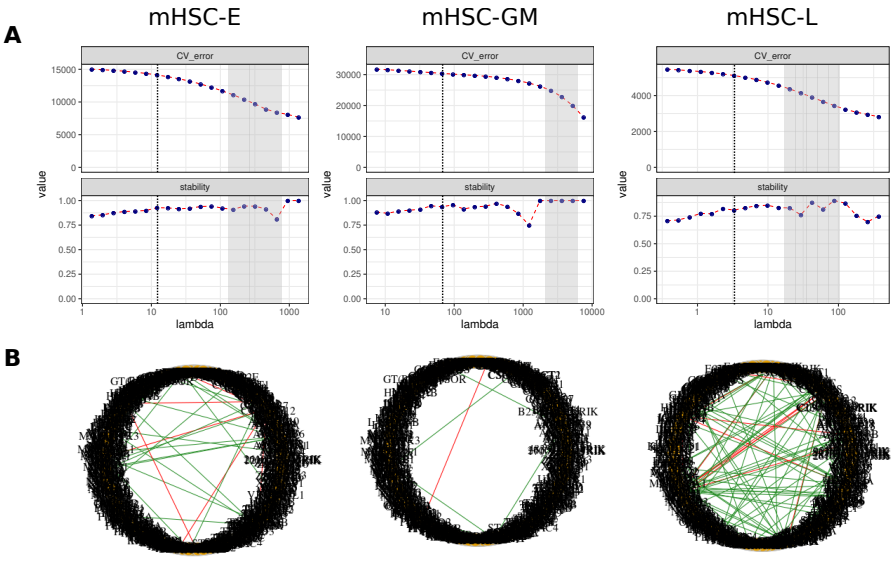


Figure A1: Alternative GRNs using data imputation with dca.
(A) Model selection criteria on imputed data similar to Fig. 5.21A using alternative λ values (dotted line) (B) Alternative GRNs compared to Fig. 5.21.

A.3 EFFECT OF IMPUTATION ON GENE REGULATORY NETWORK RECONSTRUCTION

A.3.1 Data collection and preprocessing of scRNAseq data

We collected preprocessed and normalized experimental scRNAseq count data provided in the BEELINE paper Pratapa et al., 2020. Here, the authors also provide the corresponding pseudotime for each dataset / cell type. Please refer to the BEELINE paper for information about preprocessing, normalization, and pseudotime inference. However, dca needs unnormalized raw count data. Therefore, we downloaded the fastq files using the corresponding accession numbers: GSE75748 (hESC) (Chu et al., 2016), GSE81252 (hHEP) (Camp et al., 2017), GSE98664 (mESC) (Hayashi et al., 2018), GSE48968 (mDC) (Shalek et al., 2014) and GSE81682 (mHSC) (Nestorowa et al., 2016). For human and mouse we aligned the fastq files to hg19 (GENCODE release 29) or mm10 (GENCODE release M19), respectively and counted the reads per gene using STAR (version 2.7.4a) (Dobin et al., 2013). Following the BEELINE approach, using normalized count data we select the top 500 most variable genes (or top 1000 most variable genes respectively) across pseudotime using a general additive model ('gam' R package). In addition to these genes we also include significantly varying TFs (Bonferroni corrected p-value < 0.01). We filter both imputed and unimputed scRNAseq data using the same set of (i) top 500 most variable genes (or top 1000 HVGs) and (ii) all significantly varying TFs, in order to make a fair comparison between networks inferred using imputed and unimputed data.

A.3.2 Code availability

All relevant scripts and R notebooks for reproducing the results are available at Github (`https://github.com/lylamha/imputation_GRN_inference`). The release includes tutorials from data imputation to the evaluation of the reconstructed networks. It covers the evaluation pipeline with the corresponding analyses and plotting results.

A.3.3 Imputation

To impute scRNAseq data we use `dca` (version 0.2.3), `knnsmooth` (version 2.1), `magic` ('`Rmagic`' R package version 2.0.3) and `saver` ('SAVER' R package version 1.1.2). Our rationale for selecting knnsmooth, magic and saver is based on a previous comprehensive benchmark evaluation of various imputation methods (Hou et al., 2020). Additionally, we also include dca as it has been explicitly recommended as improving GRN reconstruction (Eraslan et al., 2019). We apply each imputation method to normalized count data except for dca where we use the raw counts (see github page).

A.3.4 Network reconstruction via BEELINE

Several tools have been developed to infer GRNs from scRNAseq data differing in their algorithmic approach. They can be categorized into four main classes: correlation-, regression-, mutual information- or modelling-based approaches (Pratapa et al., 2020). In this study we evaluated PIDC, GENIE3 and GRNBoost2 which have been previously recommended by Pratapa et al., 2020. Moreover, we included PPCOR as a partial regression based algorithm providing a baseline GRN algorithm. We use the imputed and unimputed scRNAseq data as input matrices for network reconstruction with PIDC, GENIE3 and GRNBoost2 using default parameters. To this end, we use the evaluation framework BEELINE (version 1.0). In order to evaluate PPCOR results we adjusted the code of the BEELINE framework. In the publicly available version of BEELINE PPCOR considers the corrected p-values of each reported interaction with its respective partial correlation value. However, in our case there were only NA's produced due to ill-conditioned matrices. Thus, we discard the p-values and use a threshold of 0.1 absolute partial correlation value and selected interactions higher than the threshold.
As part of the BEELINE pipeline we first run `'BLRunner.py'` to reconstruct the networks. Network reconstruction methods may compute undirected or directed edges, while the STRING database contains undirected edges. Thus, in evaluating a network reconstruction method that predicts undirected edges, for both STRING and predicted networks undirected edges get substituted by two opposing directed edges. For the comparison to the evaluation networks, we only consider and filter for edges going out of TFs. With this convention, bidirectional edges get counted only once (except where two TFs are connected by an interaction). This is meant to minimize the advantage which a method producing undirected edges might possibly have.

Finally, we use `'BLevaluater.py'` to compute early precision scores evaluating the performance of each network by comparing it to an evaluation network. Here, we choose the functional protein-protein interaction database STRING and cell-type specific ChIPseq derived network provided by the BEELINE framework. We filter the network genes that only occur in the input expression matrix.
By using early precision scores we only analyze the top-k networks.

A.3.5 Characterizing the reconstructed networks

Top-k network. For comparability reasons we focus our analyses on the top-k networks. The top-k network of a reconstructed network includes the first k interactions selected by their ranks which were assigned by descendingly ordered edge weights. Here, k represents the number of positive interactions in the evaluation network. Interactions can share the same ranks, e.g., the forward and backward interactions in an undirected graph. So with k interactions reported in the evaluation network we select all interactions whose ranks are lower than or equal to k obtaining the top-k network. Note, that the number of reported interactions can be higher than k.

Network density and node degree. Taking into account the interaction between transcription factors and genes only the network density is calculated by $numEdges/((numGenes \cdot numTFs)$ –

numTFs). In order to calculate the node degree we consider all out- and incoming edges for a given node.

A.3.6 Methodology of evaluation

Early Precision Ratios (EPR). Most of the network reconstruction algorithms infer networks that are close to a full graph. Being more interested in the most important predicted interactions we analyzed the interactions with the highest (absolute) weights and thus the top ranked interactions within the network (top-k network). For this reason and to evaluate the performance of each inferred network based on using early precision sores (EP) which is given by the number of TP divided by the number of positively predicted observations within the top-k network. EP scores were calculated using BEELINE. Each dataset has a different underlying evaluation subnetwork, hence different evaluations regarding the random predictor. To account for these differences and in order to maintain comparability across datasets we divide the EP scores by the network density (see formula above) of each evaluation subnetwork obtaining EP ratios (EPR). Thus, EPR of 1 is indicative of a random predictor in all experimental datasets. To compare the performance of network inference in each imputation method with the corresponding unimputed data, we calculate log2-ratios between $\text{EPR}_{imputed}$ and $\text{EPR}_{unimputed}$.

ANOVA. For each dataset we performed a separate two-way ANOVA using the built-in R function: First, we fit a linear model using the log2-ratios of EPR values as a target variable and both, the GRN algorithm and the imputation as regressor variables. Secondly, we summarize the variance model of the linear fit and report the p-values. The source code is included on the github page.

Network similarities. In order to compare similarities across the reconstructed networks we select the top 500 interactions reported in each model. Given two networks, similarity scores are obtained by the Jaccard index which is defined as the number of overlapping interactions divided by the number of unified reported interactions. Repeating this in a pairwise iterative manner we obtain a similarity matrix which we use as an input for a heatmap that is clustered row- and column-wise ('pheatmap' R Package). We calculate adjusted rand index (ARI) scores ('mclust' R package) in order to evaluate the clustering results based on an annotation label (Hubert et al., 1985). As annotation labels we use the network reconstruction algorithm as well as the imputation method. We compare ARI scores across datasets obtained by the two labels using the pairwise wilcoxon rank sum test.

A.3.7 Supplementary Information

Table A1: Analysis of variance of performance scores for each dataset. ANOVA results on EPR log-fold-ratios. Significance codes are 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 0. Higher significance p-values in imputation give evidence that a higher variance within imputation methods compared to GRN algorithms is prevalent, and vice versa. .

data	factor	sum of squares	mean of squares	p-values
hESC	GRN	0.0455	0.0228	0.68436
	imputation	2.3036	0.7679	0.00435 **
hHep	GRN	0.0205	0.0103	0.75904
	imputation	1.4728	0.4909	0.00421 **
mDC	GRN	0.0061	0.00307	0.68601
	imputation	0.3728	0.12428	0.00275 **
mESC	GRN	0.1324	0.0662	0.00638 **
	imputation	1.1478	0.3826	3.64e-05 ***
mHSC-E	GRN	0.1456	0.07281	0.07206 .
	imputation	0.6497	0.21657	0.00541 **
mHSC-GM	GRN	0.1748	0.08739	0.000247 ***
	imputation	0.5445	0.18151	2.02e-05 ***
mHSC-L	GRN	0.11682	0.05841	0.000572 ***
	imputation	0.08218	0.02739	0.003099 **

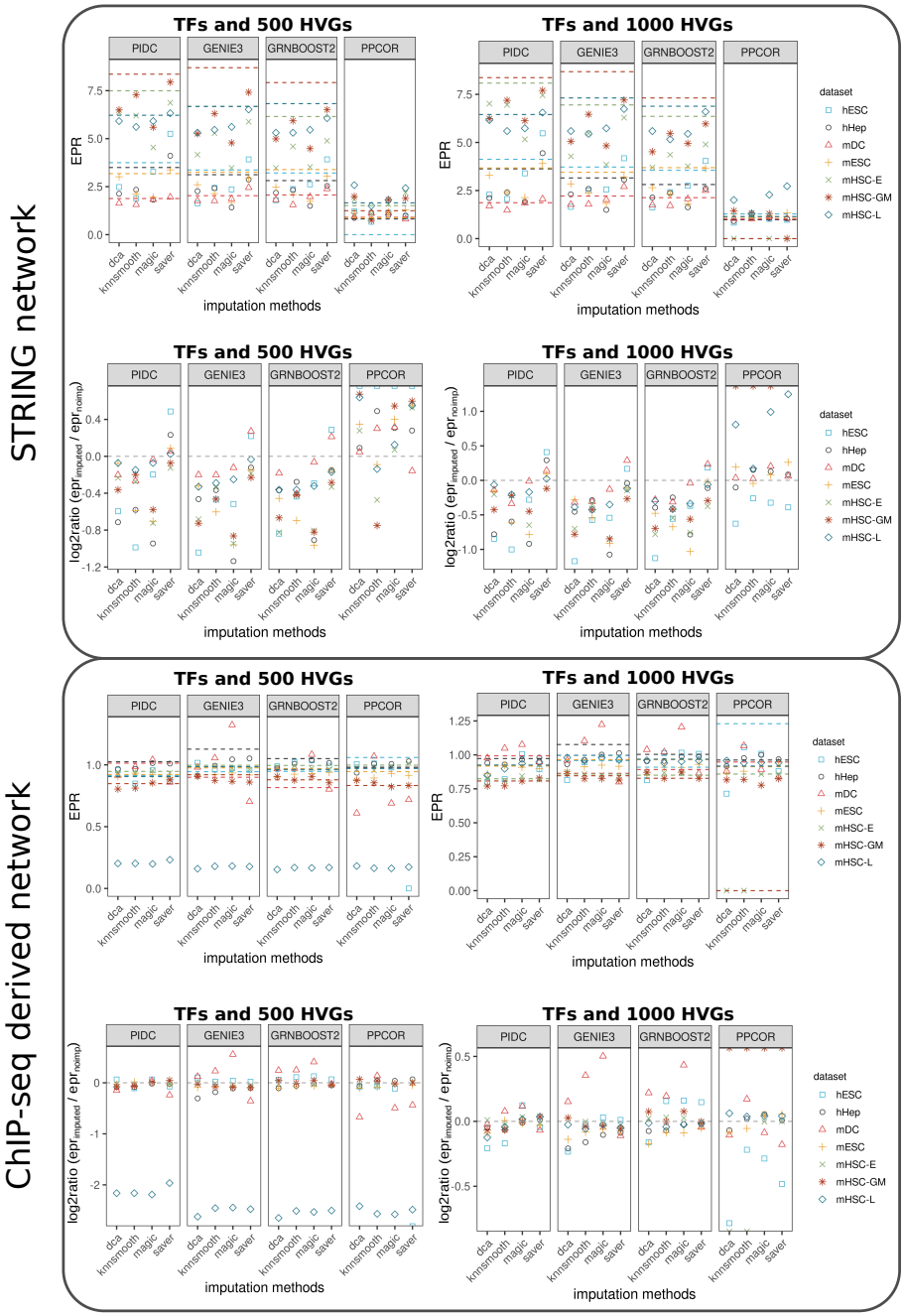


Figure A2: Performance scores of network models including PPCOR compared to STRING and ChIP-seq derived networks as evaluation networks. Upper panel is similar to Fig. 6.2 comparing the inferred network results with the STRING network. Here, we include PPCOR as a baseline GRN algorithm. PPCOR performs almost similarly to a random predictor. In some cases PPCOR failed to run due to ill-conditioned data matrices corresponding to EPR scores equal to 0. Below panel compares the inferred network results with cell type-specific ChIP-seq derived networks. In both prefiltered datasets the performances are close to random. Imputation does not improve the network predictions. Due to normalization by using the network density, the EPR scores in mHSC-L imputed data differ strongly from the unimputed data. Here, low numbers of genes/ TFs and edges lead to different network densities. In both evaluation scenarios the results between the prefiltered datasets based on the number of highly variable genes (HVGs) are comparable. Hence, varying the number of genes has little effect on the network performance predictions.

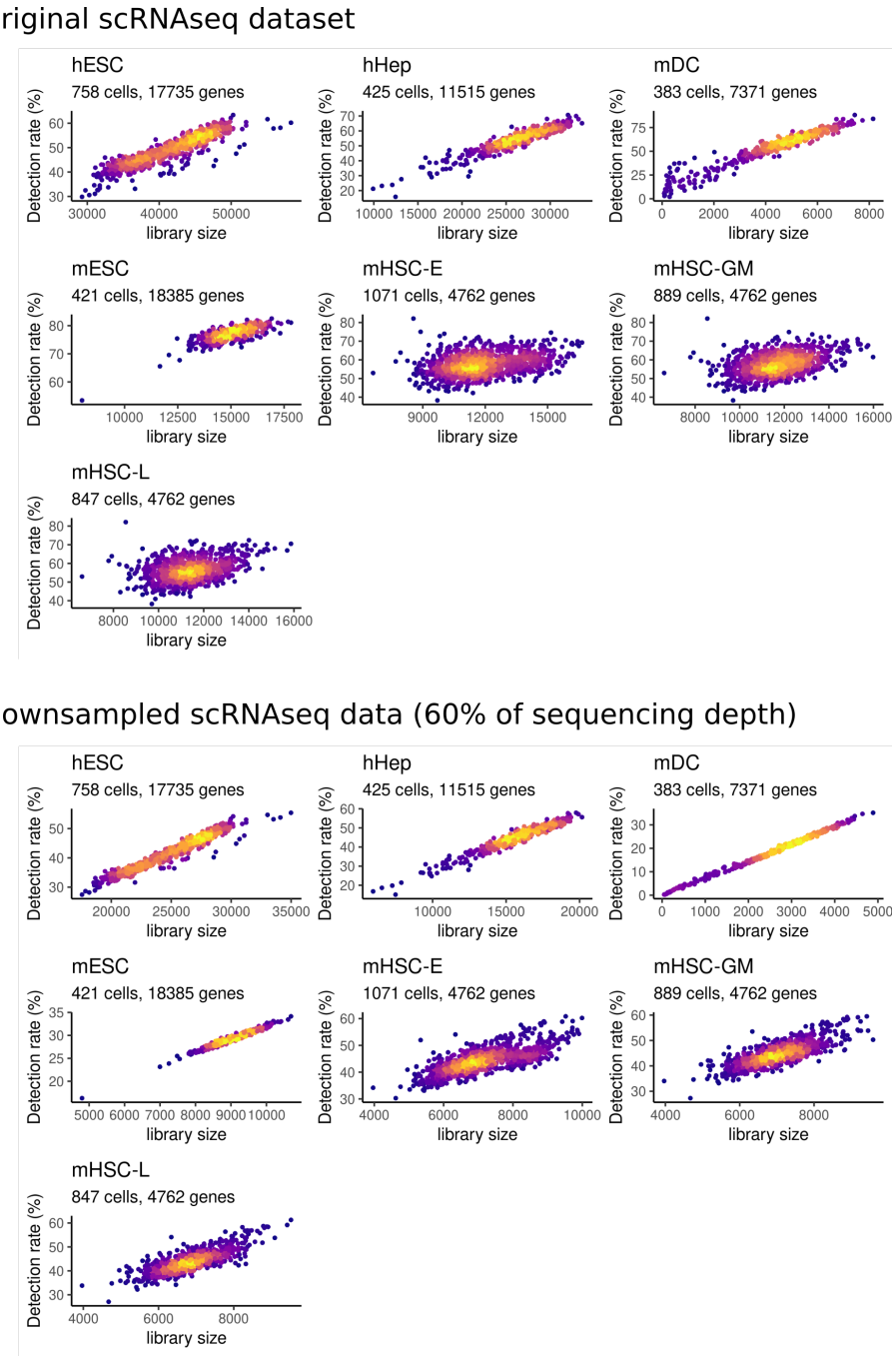


Figure A3: Gene detection rate and library size in experimental scRNAseq datasets (original and downsampled).
Scatterplots colored by density of points (cells). Gene detection using a threshold of gene count > 0. Library size determined by the sum of all gene counts. Downsampling procedure performed by sampling n times (60% of the original library size) according to the multinomial distribution.

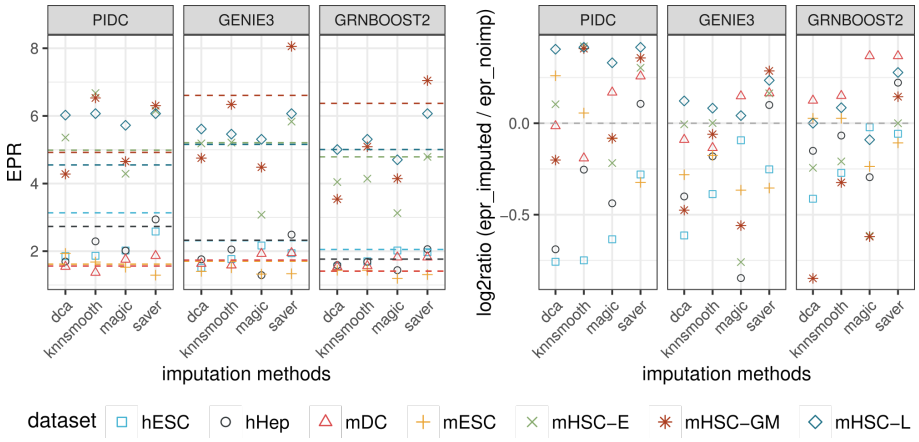


Figure A4: Performance measures of network models obtained by downsampled dataset.
Performance scores reported on downsampled scRNAseq data (60% of original library size) and prefiltered dataset (TFs and 500 HVGs). (Left) Absolute EPR scores. Dashed line represents EPR scores obtained without imputation. (Right) log2-ratios between imputed and unimputed EPR scores. Log2ratio = 0 represents no change in performance (grey dashed line) after imputation. Generally, more improvements (positive log2ratios) than in the respective column of Figure 6.2 (TFs and 500 HVGs).

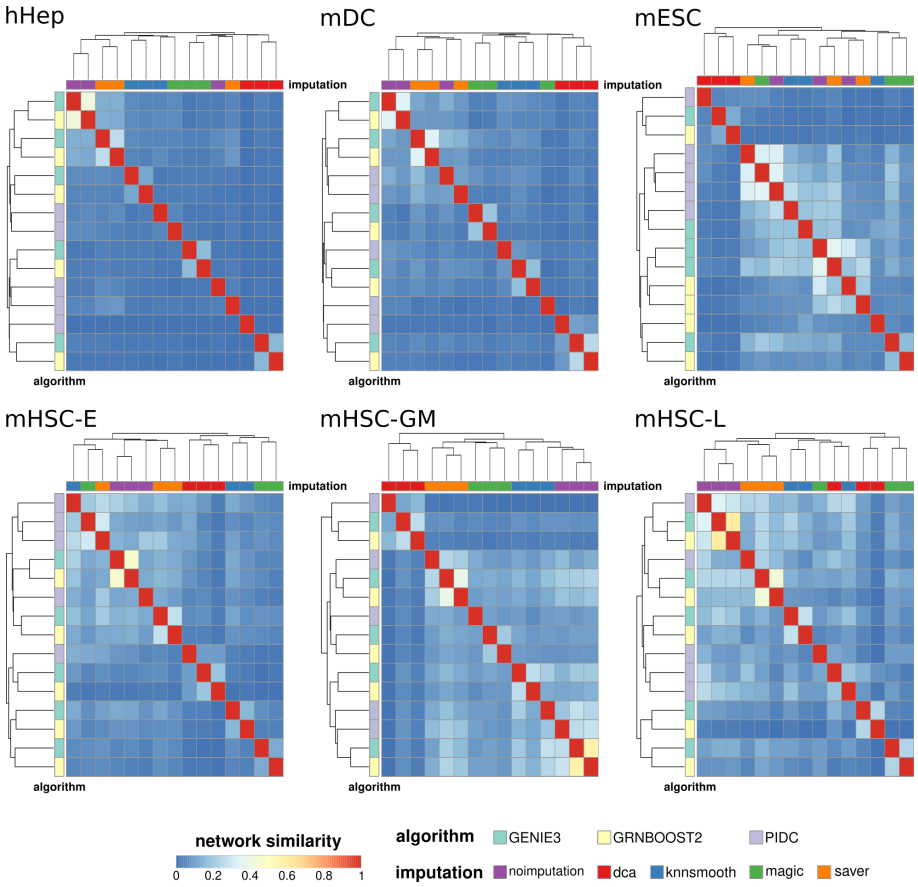


Figure A5: Network similarities across all models and cell types. According to Figure 6.3B we inspect the heatmap of network similarities of the remaining cell types. Network similarity scores obtained by pairwise Jaccard index from top500 interactions. Columns are annotated by imputation method, rows are annotated by GRN reconstruction algorithm.

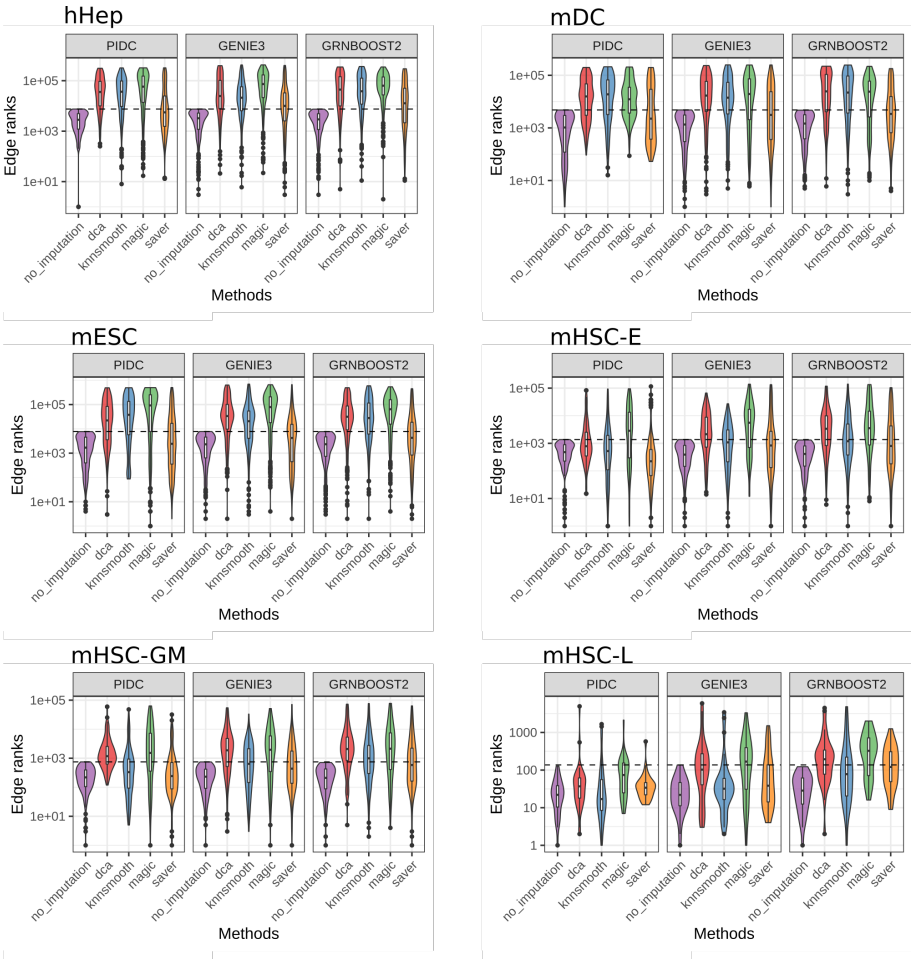


Figure A6: True positive interactions identified on unimputed data and their change in edge ranks after imputation.
According to Figure 6.6A we inspect the change of unimputed TP ranks after imputation in the remaining cell types. Corrected p-values obtained by Wilcoxon rank sum test can be taken from Suppl. Tab. A2.

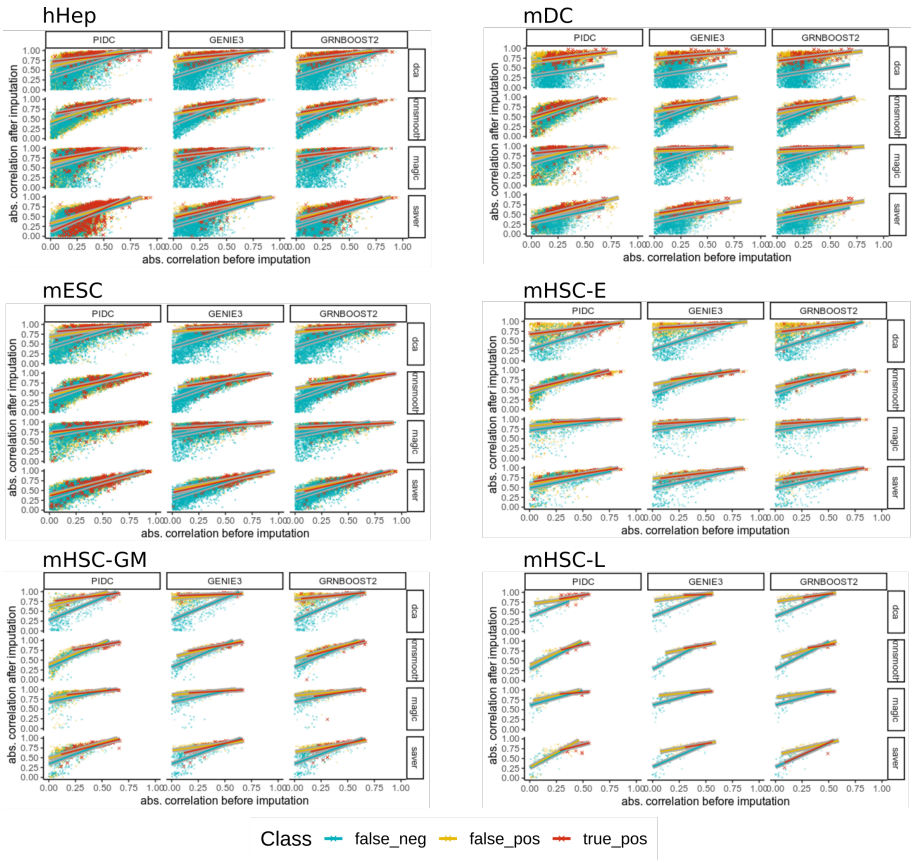


Figure A7: Absolute Pearson's correlation coefficients before and after imputation colored by prediction class obtained in each model.
According to Figure 6.6B we inspect the change of correlation values for TP, FP and FN classified by each model in the remaining cell types. Colors correspond to the prediction classes in each model.

Table A2: Differences in TP edge rank distribution before and after imputation. Corrected p-values (Bonferroni method) obtained after Wilcoxon rank sum test between ranks of unimputed true positive edges and their respective ranks after imputation.

data	imputation	PIDC	GENIE3	GRNBoost2
hESC	dca	6.23E-59	1.35E-52	3.22E-57
	knnsmooth	9.30E-70	5.56E-30	1.72E-35
	magic	5.41E-45	1.89E-47	1.68E-51
	saver	1.99E-15	5.46E-12	8.93E-13
hHep	dca	2.77E-144	3.67E-89	5.47E-83
	knnsmooth	1.11E-130	3.88E-85	5.87E-86
	magic	1.26E-126	1.19E-125	1.86E-108
	saver	1.66E-23	1.08E-37	1.34E-29
mDC	dca	1.15E-48	1.91E-39	8.93E-42
	knnsmooth	1.47E-46	5.11E-38	3.20E-38
	magic	2.14E-58	6.35E-31	9.12E-32
	saver	2.69E-10	7.78E-08	3.66E-10
mESC	dca	2.85E-75	6.53E-79	7.09E-83
	knnsmooth	2.84E-75	9.74E-59	5.33E-77
	magic	1.54E-78	1.25E-80	4.75E-90
	saver	3.61E-05	3.28E-07	7.37E-10
mHSC-E	dca	1.93E-08	2.97E-24	6.55E-25
	knnsmooth	1	4.70E-07	1.72E-10
	magic	2.44E-14	6.80E-27	4.39E-26
	saver	1	0.001687	0.0004226
mHSC-GM	dca	5.32E-44	1.02E-33	7.75E-32
	knnsmooth	0.05846	2.32E-09	9.27E-17
	magic	8.66E-21	2.31E-23	2.89E-22
	saver	1	8.45E-07	6.18E-09
mHSC-L	dca	1	0.006637	1.51E-05
	knnsmooth	1	1	0.3427
	magic	0.005388	0.004132	3.54E-06
	saver	0.7522	1	0.0006845

BIBLIOGRAPHY

Alberts, Bruce, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter Walter (Dec. 2014). *Molecular Biology Of The Cell*. 6th ed. New York, NY: Garland Science. ISBN: 0815344643.

Anders, Simon and Wolfgang Huber (Oct. 2010). "Differential expression analysis for sequence count data." In: *Genome Biology* 11.10, R106. ISSN: 1465-6914. DOI: 10.1186/gb-2010-11-10-r106.

Andrews, Tallulah S and Martin Hemberg (Nov. 2018). "False signals induced by single-cell imputation." In: *F1000Research* 7, p. 1740. DOI: 10.12688/f1000research.16613.2.

Angerer, Philipp, Lukas Simon, Sophie Tritschler, F. Alexander Wolf, David Fischer, and Fabian J. Theis (Aug. 2017). "Single cells make big data: New challenges and opportunities in transcriptomics." In: *Current Opinion in Systems Biology* 4, pp. 85–91. ISSN: 24523100. DOI: 10.1016/j.coisb.2017.07.004.

Baran, Yael, Akhiad Bercovich, Arnau Sebe-Pedros, Yaniv Lubling, Amir Giladi, Elad Chomsky, Zohar Meir, Michael Hoichman, Aviezer Lifshitz, and Amos Tanay (Oct. 2019). "MetaCell: analysis of single-cell RNA-seq data using K-nn graph partitions." In: *Genome Biology* 20.1, p. 206. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1812-2.

Bentley, David R et al. (Nov. 2008). "Accurate whole human genome sequencing using reversible terminator chemistry." In: *Nature* 456.7218, pp. 53–59. DOI: 10.1038/nature07517.

Berge, Koen Van den, Fanny Perraudeau, Charlotte Soneson, Michael I Love, Davide Risso, Jean-Philippe Vert, Mark D Robinson, Sandrine Dudoit, and Lieven Clement (Feb. 2018). "Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications." In: *Genome Biology* 19.1, p. 24. ISSN: 1474-760X. DOI: 10.1186/s13059-018-1406-4.

Blencowe, Montgomery, Douglas Arneson, Jessica Ding, Yen-Wei Chen, Zara Saleem, and Xia Yang (July 2019). "Network modeling of single-cell omics data: challenges, opportunities, and progresses." In: *Emerging topics in life sciences* 3.4, pp. 379–398. ISSN: 2397-8554. DOI: 10.1042/etls20180176.

Brawand, David et al. (Oct. 2011). "The evolution of gene expression levels in mammalian organs." In: *Nature* 478.7369, pp. 343–348. ISSN: 1476-4687. DOI: 10.1038/nature10532.

Breda, Jeremie, Mihaela Zavolan, and Erik J van Nimwegen (Dec. 2019). "Bayesian inference of the gene expression states of single cells from scRNA-seq data." In: *BioRxiv*. DOI: 10.1101/2019.12.28.889956.

Brennecke, Philip et al. (Nov. 2013). "Accounting for technical noise in single-cell RNA-seq experiments." In: *Nature Methods* 10.11, pp. 1093–1095. DOI: 10.1038/nmeth.2645.

Buettner, Florian, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle (Feb. 2015). "Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells." In: *Nature Biotechnology* 33.2, pp. 155–160. DOI: 10.1038/nbt.3102.

Camp, J Gray et al. (Dec. 2015). "Human cerebral organoids recapitulate gene expression programs of fetal neocortex development." In: *Proceedings of the National Academy of Sciences of the United States of America* 112.51, pp. 15672–15677. DOI: 10.1073/pnas.1520760112.

Camp, J Gray et al. (June 2017). "Multilineage communication regulates human liver bud development from pluripotency." In: *Nature* 546.7659, pp. 533–538. ISSN: 0028-0836. DOI: 10.1038/nature22796.

Chan, Thalia E, Michael P H Stumpf, and Ann C Babtie (Sept. 2017). "Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures." In: *Cell Systems* 5.3, 251–267.e3. DOI: 10.1016/j.cels.2017.08.014.

Chen, Mengjie and Xiang Zhou (Nov. 2018a). "VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies." In: *Genome Biology* 19.1, p. 196. DOI: 10.1186/s13059-018-1575-1.

Chen, Shuonan and Jessica C Mar (June 2018b). "Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data." In: *BMC Bioinformatics* 19.1, p. 232. DOI: 10.1186/s12859-018-2217-z.

Chu, Li-Fang, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A Thomson (Aug. 2016). "Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm." In: *Genome Biology* 17.1, p. 173. DOI: 10.1186/s13059-016-1033-x.

Clevers, H et al. (Mar. 2017). "What is your conceptual definition of "cell type" in the context of a mature organism?" In: *Cell Systems* 4.3, pp. 255–259. DOI: 10.1016/j.cels.2017.03.006.

Collins, Francis S, Michael Morgan, and Aristides Patrinos (Apr. 2003). "The Human Genome Project: lessons from large-scale biology." In: *Science* 300.5617, pp. 286–290. DOI: 10.1126/science.1084564.

Consortium, GTEx (May 2015). "Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans." In: *Science* 348.6235, pp. 648–660. DOI: 10.1126/science.1262110.

Dijk, David van et al. (July 2018). "Recovering Gene Interactions from Single-Cell Data Using Data Diffusion." In: *Cell* 174.3, 716–729.e27. ISSN: 00928674. DOI: 10.1016/j.cell.2018.05.061.

Dobin, Alexander, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras (Jan. 2013). "STAR: ultrafast universal RNA-seq aligner." In: *Bioinformatics* 29.1, pp. 15–21. DOI: 10.1093/bioinformatics/bts635.

Eraslan, Gökcen, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis (Jan. 2019). "Single-cell RNA-seq denoising using a deep count autoencoder." In: *Nature Communications* 10.1, p. 390. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07931-2.

Fauci, Anthony S (Nov. 2006). "Seasonal and pandemic influenza preparedness: science and countermeasures." In: *The Journal of Infectious Diseases* 194 Suppl 2, S73–6. DOI: 10.1086/507550.

Finak, Greg et al. (Dec. 2015). "MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data." In: *Genome Biology* 16, p. 278. DOI: 10.1186/s13059-015-0844-5.

Gates, Alexander J. and Yong-Yeol Ahn (Oct. 2017). "The impact of random models on clustering similarity." In: *BioRxiv*. DOI: 10.1101/196840.

Ghanbari, Mahsa, Julia Lasserre, and Martin Vingron (Mar. 2019). "The Distance Precision Matrix: computing networks from non-linear relationships." In: *Bioinformatics* 35.6, pp. 1009–1017. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty724.

Haghverdi, Laleh, Florian Buettner, and Fabian J Theis (Sept. 2015). "Diffusion maps for high-dimensional single-cell analysis of differentiation data." In: *Bioinformatics* 31.18, pp. 2989–2998. DOI: 10.1093/bioinformatics/btv325.

Han, Xiaoping et al. (Feb. 2018). "Mapping the Mouse Cell Atlas by Microwell-Seq." In: *Cell* 172.5, 1091–1107.e17. DOI: 10.1016/j.cell.2018.02.001.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. 2nd. Springer Series in Statistics. New York, NY: Springer New York, pp. 106–119. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7.

Hayashi, Tetsutaro, Haruka Ozaki, Yohei Sasagawa, Mana Umeda, Hiroki Danno, and Itoshi Nikaido (Feb. 2018). "Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs." In: *Nature Communications* 9.1, p. 619. DOI: 10.1038/s41467-018-02866-0.

Heimberg, Graham, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson (Apr. 2016). "Low Dimensionality in Gene Expression Data Enables the Accurate Extraction of Transcriptional Programs from Shallow Sequencing." In: *Cell Systems* 2.4, pp. 239–250. ISSN: 24054712. DOI: 10.1016/j.cels.2016.04.001.

Heldt, Frank S, Sascha Y Kupke, Sebastian Dorl, Udo Reichl, and Timo Frensing (Nov. 2015). "Single-cell analysis and stochastic modelling unveil large cell-to-cell variability in influenza A virus infection." In: *Nature Communications* 6, p. 8938. DOI: 10.1038/ncomms9938.

Hicks, Stephanie C, F William Townes, Mingxiang Teng, and Rafael A Irizarry (Oct. 2018). "Missing data and technical variability in single-cell RNA-sequencing experiments." In: *Biostatistics* 19.4, pp. 562–578. DOI: 10.1093/biostatistics/kxx053.

Hou, Wenpin, Zhicheng Ji, Hongkai Ji, and Stephanie C Hicks (Aug. 2020). "A systematic evaluation of single-cell RNA-sequencing imputation methods." In: *Genome Biology* 21.1, p. 218. ISSN: 1474-760X. DOI: 10.1186/s13059-020-02132-x.

Huang, Mo, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang (June 2018). "SAVER: gene expression recovery for single-cell RNA sequencing." In: *Nature Methods* 15.7, pp. 539–542. ISSN: 1548-7091. DOI: 10.1038/s41592-018-0033-z.

Hubert, Lawrence and Phipps Arabie (Dec. 1985). "Comparing partitions." In: *Journal of Classification* 2.1, pp. 193–218. ISSN: 0176-4268. DOI: 10.1007/{BF01908075}.

Huynh-Thu, Vân Anh, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts (Sept. 2010). "Inferring regulatory networks from expression data using tree-based methods." In: *Plos One* 5.9. DOI: 10.1371/journal.pone.0012776.

Ilicic, Tomislav, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Frederik Otzen Bagger, Davis James McCarthy, John C Marioni, and Sarah A Teichmann (Feb. 2016). "Classification of low quality cells from single-cell RNA-seq data." In: *Genome Biology* 17, p. 29. DOI: 10.1186/s13059-016-0888-1.

Islam, Saiful, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson (July 2011). "Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq." In: *Genome Research* 21.7, pp. 1160–1167. ISSN: 1549-5469. DOI: 10.1101/gr.110882.110.

Iwasaki, Hiromi and Koichi Akashi (June 2007). "Myeloid lineage commitment from the hematopoietic stem cell." In: *Immunity* 26.6, pp. 726–740. DOI: 10.1016/j.immuni.2007.06.004.

Jiang, Lichun, Felix Schlesinger, Carrie A Davis, Yu Zhang, Renhua Li, Marc Salit, Thomas R Gingeras, and Brian Oliver (Sept. 2011). "Synthetic spike-in standards for RNA-seq experiments." In: *Genome Research* 21.9, pp. 1543–1551. DOI: 10.1101/gr.121095.111.

Jiang, Ruochen, Tianyi Sun, Dongyuan Song, and Jingyi Jessica Li (Jan. 2022). "Statistics or biology: the zero-inflation controversy about scRNA-seq data." In: *Genome Biology* 23.1, p. 31. ISSN: 1474-760X. DOI: 10.1186/s13059-022-02601-5.

Johnson, W Evan, Cheng Li, and Ariel Rabinovic (Jan. 2007). "Adjusting batch effects in microarray expression data using empirical Bayes methods." In: *Biostatistics* 8.1, pp. 118–127. DOI: 10.1093/biostatistics/kxj037.

Kharchenko, Peter V, Lev Silberstein, and David T Scadden (July 2014). "Bayesian approach to single-cell differential expression analysis." In: *Nature Methods* 11.7, pp. 740–742. DOI: 10.1038/nmeth.2967.

Kim, Chanwoo, Hanbin Lee, Juhee Jung, Keehoon Jung, and Buhm Han (Nov. 2020). "MarcoPolo: a clustering-free approach to the exploration of differentially expressed genes along with group information in single-cell RNA-seq data." In: *BioRxiv*. DOI: 10.1101/2020.11.23.393900.

Kim, Seongho (Nov. 2015). "ppcor: An R package for a fast calculation to semi-partial correlation coefficients." In: *Communications for statistical applications and methods* 22.6, pp. 665–674. DOI: 10.5351/{CSAM}.2015.22.6.665.

Klein, Allon M, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner (May 2015). "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells." In: *Cell* 161.5, pp. 1187–1201. DOI: 10.1016/j.cell.2015.04.044.

Kobak, Dmitry and Philipp Berens (Nov. 2019). "The art of using t-SNE for single-cell transcriptomics." In: *Nature Communications* 10.1, p. 5416. ISSN: 2041-1723. DOI: 10.1038/s41467-019-13056-x.

Kolodziejczyk, Aleksandra A et al. (Oct. 2015a). "Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation." In: *Cell Stem Cell* 17.4, pp. 471–485. DOI: 10.1016/j.stem.2015.09.011.

Kolodziejczyk, Aleksandra A, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann (May 2015b). "The technology and biology of single-cell RNA sequencing." In: *Molecular Cell* 58.4, pp. 610–620. DOI: 10.1016/j.molcel.2015.04.005.

Krumsiek, Jan, Carsten Marr, Timm Schroeder, and Fabian J Theis (Aug. 2011). "Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network." In: *Plos One* 6.8, e22649. DOI: 10.1371/journal.pone.0022649.

Kupke, Sascha Young, Lam-Ha Ly, Stefan Thomas Börno, Alexander Ruff, Bernd Timmermann, Martin Vingron, Stefan Haas, and Udo Reichl (Jan. 2020). "Single-Cell Analysis Uncovers a Vast Diversity in Intracellular Viral Defective Interfering RNA Content Affecting the Large Cell-to-Cell Heterogeneity in Influenza A Virus Replication." In: *Viruses* 12.1. DOI: 10.3390/v12010071.

Laiosa, Catherine V, Matthias Stadtfeld, and Thomas Graf (2006). "Determinants of lymphoid-myeloid lineage diversification." In: *Annual Review of Immunology* 24, pp. 705–738. DOI: 10.1146/annurev.immunol.24.021605.090742.

Linderman, George C., Jun Zhao, and Yuval Kluger (Aug. 2018). "Zero-preserving imputation of scRNA-seq data using low-rank approximation." In: *BioRxiv*. DOI: 10.1101/397588.

Lockhart, David J. et al. (Dec. 1996). "Expression monitoring by hybridization to high-density oligonucleotide arrays." In: *Nature Biotechnology* 14 (13). ISSN: 1087-0156. DOI: 10.1038/nbt1296-1675.

Lopez, Romain, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef (Nov. 2018). "Deep generative modeling for single-cell transcriptomics." In: *Nature Methods* 15.12, pp. 1053–1058. ISSN: 1548-7091. DOI: 10.1038/s41592-018-0229-2.

Luecken, Malte D and Fabian J Theis (June 2019). "Current best practices in single-cell RNA-seq analysis: a tutorial." In: *Molecular Systems Biology* 15.6, e8746. ISSN: 1744-4292. DOI: 10.15252/msb.20188746.

Lun, Aaron T L, Karsten Bach, and John C Marioni (Apr. 2016). "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts." In: *Genome Biology* 17, p. 75. DOI: 10.1186/s13059-016-0947-7.

Ly, Lam-Ha and Martin Vingron (Dec. 2021). "Effect of imputation on gene network reconstruction from single-cell RNA-seq data." In: *Patterns*, p. 100414. ISSN: 26663899. DOI: 10.1016/j.patter.2021.100414.

Maaten, L Van der and G Hinton (2008). "Visualizing data using t-SNE." In: *Journal of machine learning research*.

Macosko, Evan Z et al. (May 2015). "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets." In: *Cell* 161.5, pp. 1202–1214. ISSN: 00928674. DOI: 10.1016/j.cell.2015.05.002.

McDowell, D G, N A Burns, and H C Parkes (July 1998). "Localised sequence regions possessing high melting temperatures prevent the amplification of a DNA mimic in competitive PCR." In: *Nucleic Acids Research* 26.14, pp. 3340–3347. DOI: 10.1093/nar/26.14.3340.

McInnes, L, J Healy, and J Melville (2018). "Umap: Uniform manifold approximation and projection for dimension reduction." In: *arXiv preprint arXiv:1802.03426*.

Meinshausen, Nicolai and Peter Bühlmann (June 2006). "High-dimensional graphs and variable selection with the Lasso." In: *The Annals of Statistics* 34.3, pp. 1436–1462. ISSN: 0090-5364. DOI: 10.1214/009053606000000281.

Moerman, Thomas, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts (June 2019). "GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks." In: *Bioinformatics* 35.12, pp. 2159–2161. DOI: 10.1093/bioinformatics/bty916.

Mongia, Aanchal, Debarka Sengupta, and Angshul Majumdar (Jan. 2019). "McImpute: Matrix Completion Based Imputation for Single Cell RNA-seq Data." In: *Frontiers in genetics* 10, p. 9. ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00009.

Mortazavi, Ali, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold (July 2008). "Mapping and quantifying mammalian transcriptomes by RNA-Seq." In: *Nature Methods* 5.7, pp. 621–628. ISSN: 1548-7105. DOI: 10.1038/nmeth.1226.

Mulekar, Madhuri S. and C. Scott Brown (2017). "Distance and similarity measures." In: *Encyclopedia of social network analysis and mining*. Ed. by Reda Alhajj and Jon Rokne. New York, NY: Springer New York, pp. 1–16. ISBN: 978-1-4614-7163-9. DOI: 10.1007/978-1-4614-7163-9_141-1.

Nestorowa, Sonia, Fiona K Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola K Wilson, David G Kent, and Berthold Göttgens (Aug. 2016). "A single-cell

resolution map of mouse hematopoietic stem and progenitor cell differentiation." In: *Blood* 128.8, e20–31. DOI: 10.1182/blood-2016-05-716480.

Papili Gao, Nan, S M Minhaz Ud-Dean, Olivier Gandrillon, and Rudiyanto Gunawan (Jan. 2018). "SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles." In: *Bioinformatics* 34.2, pp. 258–266. DOI: 10.1093/bioinformatics/btx575.

Patro, Rob, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford (Apr. 2017). "Salmon provides fast and bias-aware quantification of transcript expression." In: *Nature Methods* 14.4, pp. 417–419. DOI: 10.1038/nmeth.4197.

Pearson, Karl (1901). "LIII. On lines and planes of closest fit to systems of points in space." In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572. DOI: 10.1080/14786440109462720.

Picelli, Simone, Omid R Faridani, Asa K Björklund, Gösta Winberg, Sven Sagasser, and Rickard Sandberg (Jan. 2014). "Full-length RNA-seq from single cells using Smart-seq2." In: *Nature Protocols* 9.1, pp. 171–181. DOI: 10.1038/nprot.2014.006.

Pratapa, Aditya, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and T M Murali (Jan. 2020). "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data." In: *Nature Methods* 17.2, pp. 147–154. ISSN: 1548-7091. DOI: 10.1038/s41592-019-0690-6.

Ramsköld, Daniel et al. (Aug. 2012). "Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells." In: *Nature Biotechnology* 30.8, pp. 777–782. DOI: 10.1038/nbt.2282.

Reichard, Andrew and Kewal Asosingh (2019). "Best Practices for Preparing a Single Cell Suspension from Solid Tissues for Flow Cytometry." In: *Cytometry. Part A: the Journal of the International Society for Analytical Cytology* 95.2, pp. 219–226. DOI: 10.1002/cyto.a.23690.

Risso, Davide, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert (Jan. 2018). "A general and flexible method for signal extraction from single-cell RNA-seq data." In: *Nature Communications* 9.1, p. 284. DOI: 10.1038/s41467-017-02554-5.

Russell, Alistair B, Elizaveta Elshina, Jacob R Kowalsky, Aartjan J W Te Velthuis, and Jesse D Bloom (July 2019). "Single-Cell Virus Sequencing of Influenza Infections That Trigger Innate Immunity." In: *Journal of Virology* 93.14. ISSN: 0022-538X. DOI: 10.1128/{JVI}.00500-19.

Schaffter, Thomas, Daniel Marbach, and Dario Floreano (Aug. 2011). "GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods." In: *Bioinformatics* 27.16, pp. 2263–2270. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btr373.

Schena, Mark, Dari Shalon, Ronald W. Davis, and Patrick O. Brown (1995). "Quantitative monitoring of gene expression patterns with a complementary DNA microarray." In: *Science*. ISSN: 00368075. DOI: 10.1126/science.270.5235.467.

Sena, Johnny A, Giulia Galotto, Nico P Devitt, Melanie C Connick, Jennifer L Jacobi, Pooja E Umale, Luis Vidali, and Callum J Bell (Sept. 2018). "Unique Molecular Identifiers reveal a novel sequencing artefact with implications for RNA-Seq based gene expression analysis." In: *Scientific Reports* 8.1, p. 13121. DOI: 10.1038/s41598-018-31064-7.

Shalek, Alex K et al. (June 2014). "Single-cell RNA-seq reveals dynamic paracrine control of cellular variation." In: *Nature* 510.7505, pp. 363–369. DOI: 10.1038/nature13437.

Shapiro, Ehud, Tamir Biezuner, and Sten Linnarsson (Sept. 2013). "Single-cell sequencing-based technologies will revolutionize whole-organism science." In: *Nature Reviews. Genetics* 14.9, pp. 618–630. DOI: 10.1038/nrg3542.

Smyth, Gordon K and Terry Speed (Dec. 2003). "Normalization of cDNA microarray data." In: *Methods* 31.4, pp. 265–273. DOI: 10.1016/s1046-2023(03)00155-5.

Soneson, Charlotte and Mark D Robinson (Feb. 2018). "Bias, robustness and scalability in single-cell differential expression analysis." In: *Nature Methods* 15.4, pp. 255–261. ISSN: 1548-7091. DOI: 10.1038/nmeth.4612.

Stegle, Oliver, Sarah A Teichmann, and John C Marioni (Mar. 2015). "Computational and analytical challenges in single-cell transcriptomics." In: *Nature Reviews. Genetics* 16.3, pp. 133–145. DOI: 10.1038/nrg3833.

Steinheuer, Lisa Maria, Sebastian Canzler, and Jörg Hackermüller (Apr. 2021). "Benchmarking scRNA-seq imputation tools with respect to network inference highlights deficits in performance at high levels of sparsity." In: *BioRxiv*. DOI: 10.1101/2021.04.02.438193.

Street, Kelly, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit (June 2018). "Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics." In: *BMC Genomics* 19.1, p. 477. DOI: 10.1186/s12864-018-4772-0.

Svensson, Valentine. *Droplet scRNA-seq is not zero inflated —What do you mean "heterogeneity"?* WEBSITE.

– (2020). "Droplet scRNA-seq is not zero-inflated." In: *Nature Biotechnology* 38.2, pp. 147–150. ISSN: 1087-0156. DOI: 10.1038/s41587-019-0379-5.

Svensson, Valentine, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah A Teichmann (Apr. 2017). "Power analysis of single-cell RNA-sequencing experiments." In: *Nature Methods* 14.4, pp. 381–387. DOI: 10.1038/nmeth.4220.

Svensson, Valentine, Roser Vento-Tormo, and Sarah A Teichmann (Mar. 2018). "Exponential scaling of single-cell RNA-seq in the past decade." In: *Nature Protocols* 13.4, pp. 599–604. ISSN: 1754-2189. DOI: 10.1038/nprot.2017.149.

Svensson, Valentine, Eduardo da Veiga Beltrame, and Lior Pachter (Nov. 2020). "A curated database reveals trends in single-cell transcriptomics." In: *Database: the Journal of Biological Databases and Curation* 2020. DOI: 10.1093/database/baaa073.

Szklarczyk, Damian et al. (Jan. 2019). "STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets." In: *Nucleic Acids Research* 47.D1, pp. D607–D613. DOI: 10.1093/nar/gky1131.

Tang, Fuchou et al. (May 2009). "mRNA-Seq whole-transcriptome analysis of a single cell." In: *Nature Methods* 6.5, pp. 377–382. ISSN: 1548-7105. DOI: 10.1038/nmeth.1315.

Tang, Fuchou, Catalin Barbacioru, Siqin Bao, Caroline Lee, Ellen Nordman, Xiaohui Wang, Kaiqin Lao, and M Azim Surani (May 2010). "Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell RNA-Seq analysis." In: *Cell Stem Cell* 6.5, pp. 468–478. DOI: 10.1016/j.stem.2010.03.015.

Tang, Fuchou, Catalin Barbacioru, Ellen Nordman, Siqin Bao, Caroline Lee, Xiaohui Wang, Brian B Tuch, Edith Heard, Kaiqin Lao, and M Azim Surani (June 2011). "Deterministic and stochastic allele specific gene expression in single mouse blastomeres." In: *Plos One* 6.6, e21208. DOI: 10.1371/journal.pone.0021208.

Tang, Wenhao, François Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Marguerat, and Vahid Shahrezaei (Feb. 2020). "bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data." In: *Bioinformatics* 36.4, pp. 1174–1181. DOI: 10.1093/bioinformatics/btz726.

Thorndike, Robert L. (Dec. 1953). "Who belongs in the family?" In: *Psychometrika* 18.4, pp. 267–276. ISSN: 0033-3123. DOI: 10.1007/{BF02289263}.

Tibshirani, Robert (Jan. 1996). "Regression shrinkage and selection via the lasso." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288. ISSN: 00359246. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

Timm, Andrea C, Jay W Warrick, and John Yin (Sept. 2017). "Quantitative profiling of innate immune activation by viral infection in single cells." In: *Integrative Biology: Quantitative Biosciences from Nano to Macro* 9.9, pp. 782–791. DOI: 10.1039/c7ib00082k.

Townes, F William, Stephanie C Hicks, Martin J Aryee, and Rafael A Irizarry (Dec. 2019). "Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model." In: *Genome Biology* 20.1, p. 295. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1861-6.

Vandenbon, Alexis and Diego Diez (Aug. 2020). "A clustering-independent method for finding differentially expressed genes in single-cell transcriptome data." In: *Nature Communications* 11.1, p. 4318. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17900-3.

Vieth, Beate, Swati Parekh, Christoph Ziegenhain, Wolfgang Enard, and Ines Hellmann (Oct. 2019). "A systematic evaluation of single cell RNA-seq analysis pipelines." In: *Nature Communications* 10.1, p. 4667. ISSN: 2041-1723. DOI: 10.1038/s41467-019-12266-7.

Wagner, Allon, Aviv Regev, and Nir Yosef (Nov. 2016). "Revealing the vectors of cellular identity with single-cell genomics." In: *Nature Biotechnology* 34.11, pp. 1145–1160. ISSN: 1087-0156. DOI: 10.1038/nbt.3711.

Wagner, Florian, Yun Yan, and Itai Yanai (Nov. 2017). "K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data." In: *BioRxiv*. DOI: 10.1101/217737.

Wolf, F Alexander, Philipp Angerer, and Fabian J Theis (Feb. 2018). "SCANPY: large-scale single-cell gene expression data analysis." In: *Genome Biology* 19.1, p. 15. DOI: 10.1186/s13059-017-1382-0.

Wong, Koon Ho, Yi Jin, and Zarmik Moqtaderi (2013). "Multiplex Illumina sequencing using DNA barcoding." In: *Current Protocols in Molecular Biology* Chapter 7, Unit 7.11. DOI: 10.1002/0471142727.mb0711s101.

Xin, Yurong et al. (Mar. 2016). "Use of the Fluidigm C1 platform for RNA sequencing of single mouse pancreatic islet cells." In: *Proceedings of the National Academy of Sciences of the United States of America* 113.12, pp. 3293–3298. DOI: 10.1073/pnas.1602306113.

Zappia, Luke, Belinda Phipson, and Alicia Oshlack (June 2018). "Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database." In: *PLoS Computational Biology* 14.6, e1006245. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006245.

Zeisel, Amit et al. (Mar. 2015). "Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq." In: *Science* 347.6226, pp. 1138–1142. DOI: 10.1126/science.aaa1934.

Zhu, Lingxue, Jing Lei, Bernie Devlin, and Kathryn Roeder (Mar. 2018). "A unified statistical framework for single cell and bulk rna sequencing data." In: *The annals of applied statistics* 12.1, pp. 609–632. ISSN: 1932-6157. DOI: 10.1214/17-{AOAS1110}.

Ziegenhain, Christoph, Beate Vieth, Swati Parekh, Björn Reinius, Amy Guillaumet-Adkins, Martha Smets, Heinrich Leonhardt, Holger Heyn, Ines Hellmann, and Wolfgang Enard (Feb. 2017). "Comparative Analysis of Single-Cell RNA Sequencing Methods." In: *Molecular Cell* 65.4, 631–643.e4. DOI: 10.1016/j.molcel.2017.01.023.

SELBSTÄNDIGKEITSERKLÄRUNG

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Lam-Ha Ly

Berlin, April 2022

Lam-Ha Ly