HelmholtzZentrum münchen
Deutsches Forschungszentrum für Gesundheit und Umwelt

Stochastic Modeling of Heterogeneous Low-Input Gene Expression: Linking Single-Cell Probability Distributions to Transcription Mechanisms

Lisa Amrhein

Juni 2021

Technische Universität München

Fakultät für Mathematik

Lehrstuhl für Mathematische Modellierung biologischer Systeme

Stochastic Modeling of Heterogeneous Low-Input Gene Expression: Linking Single-Cell Probability Distributions to Transcription Mechanisms

Lisa Amrhein

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Matthias Scherer

Prüfer der Dissertation:

1. Prof. Dr. Dr. Fabian J. Theis
2. Prof. Dr. Christiane Fuchs
3. Prof. Dr. Peter Pfaffelhuber

Die Dissertation wurde am 02.12.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 05.05.2021 angenommen.

Acknowledgments

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Doktorarbeit unterstützt und motiviert haben.
Ich danke hier vor allem meinem Doktorvater, Fabian Theis, der diese Promotion am ICB an der Schnittstelle zwischen Mathematik, Statistik und Biologie ermöglicht hat.
Ganz besonders danke ich meiner Betreuerin Christiane Fuchs für die Unterstützung und Motivation, die freie Arbeitsatmosphäre und ihren fachlichen Input.
Bedanken möchte ich mich auch bei meinen Hiwis, vor allem bei Harsha Kumar, Susanne Amrhein und Mara Santarelli, die einige wichtige Programmierarbeiten und Recherchen übernommen haben und mir oft mit ihren fachlichen Diskussionen neuen Input gegeben haben.
Ein ganz besonderer Dank gilt dem kompletten Fuchs Lab in München und Bielefeld – vor allem Hannah und Susanne – für alle tollen Unternehmungen, Unterhaltungen und auch wissenschaftliche Diskussionen. Ohne euch hätte alles nur halb so viel Spaß gemacht!
Ich möchte auch allen meinen Kollegen am ICB sowie den Mitgliedern des SFB 1243 danken. Durch die vielen Diskussionen vor allem auch auf unseren Retreats habe ich viele tolle Leute, aber auch interessante Themen kennen gelernt.
Zu guter Letzt möchte ich mich bei meinen Freunden, meiner Familie und vor allem bei Markus dafür bedanken, dass sie immer für mich da sind und mich bei allem bedingungslos unterstützen.

Abstract

Tissues are often heterogeneous in their single-cell molecular expression, and this can influence the regulation of cell fate. To understand development and disease, it is important to quantify transcriptional heterogeneity in a given tissue. Currently, this is often done with gene expression data from single-cell sequencing experiments. In addition, the joint measurement and analysis of a few cells in small-cell pools can add further interesting information that might be lost during single-cell measurements for example due to low signal.

In this thesis, I address the question of modeling low-input gene expression of single-cell measurements and such small-pool measurements based on knowledge gained from modeling single-cell data. Several tools analyze the outcome of single-cell RNA sequencing experiments, often assuming a probability distribution for the observed sequencing counts. Therefore, whenever a model-based tool is to be developed, the most appropriate discrete distribution must be determined not only in terms of model estimation, but also in terms of the interpretability, complexity and biological plausibility of inherent assumptions. To address the question of interpretability, I investigate mechanistic transcription and degradation models underlying commonly used discrete probability distributions. Well-known bottom-up approaches derive steady-state probability distributions such as Poisson or Poisson-beta distributions from different underlying transcription-degradation models (Dattani and Barahona, 2017). By turning this procedure upside down, I show how to derive a corresponding biological model from a given probability distribution. This is done via a useful connection between Ornstein-Uhlenbeck processes – a special kind of stochastic processes – and steady state probability distributions. With this I derive in one case the underlying mechanisms when using the negative binomial distribution. Realistic mechanistic models underlying this distributional assumption for mRNA counts have not yet been described explicitly but can be deducted from similar models (Berg, 1978, Paulsson et al., 2000, Raj et al., 2006). I show that the negative binomial distribution arises as steady-state distribution from a mechanistic model that produces mRNA molecules in bursts. I demonstrate empirically that using this distribution to model single-cell mRNA counts provides a convenient trade-off between computational complexity and biological simplicity. For comparison further derived distributions and mechanistic models with different inherent assumptions are included.

Furthermore, I present a model to dissect transcriptional heterogeneity from RNA sequencing counts taken from small pools of cells. In the past many tools were developed to deconvolute bulk measurements using information of purified single-cell measurements or sorted bulks (Aliee and Theis, 2020) or other prior information on the measurements (Gaujoux and Seoighe, 2013, Newman et al., 2015). However, I present the stochastic profiling algorithm which does not need any prior knowledge on the contained heterogeneity and thus does a blind deconvolution. Additionally it is not tailored to bulk samples but to measurements of small pools of cells. This algorithm was first proposed by Bajikar et al. (2014) and uses the maximum likelihood principle to parameterize heterogeneity from the cumulative continuous expression of small random pools of cells. In this thesis I provide the first complete description of this algorithm together with an evaluation of the algorithm's performance in broad simulation studies among others regarding parameter uncertainty of pool sizes. Stochastic profiling outweighs the necessary demixing of mixed samples with a saving of experimental cost and effort and less measurement errors. I generate further application opportunities in downstream analysis when investigating the inferred heterogeneity. This offers now possibilities not only to parameterize heterogeneity but also to estimate underlying pool compositions and to study detected differences between cell populations and samples.

With the advent of sequencing technologies, it has become necessary to revise the algorithm to meet the new requirements with a discrete model. Therefore, I have developed such a new version using the negative binomial distribution, so that the algorithm can now deconvolve sequencing measurements under consideration of suitable assumptions. To incorporate uncertainty of the model parameters I extended parameter inference to Bayesian methods using Hamiltonian Monte Carlo. Computational efficiency is optimized using the Stan implementation of the No-U-Turn sampler. Comparison with the standard maximum likelihood optimization shows comparable results in shorter time especially for more complex mixtures.

Finally, I show an application of the discrete algorithm to real-world data. There I study two real-world small pool datasets generated for this purpose of homogeneous mouse embryonic stem cells and presumably heterogeneous cells derived from patients suffering from acute myeloid leukemia. Although, I cannot see reduced noise or a better heterogeneity detection in relation to pool size, I observe a general gain of information with increasing pool size.

Zusammenfassung

Die Heterogenität der molekularen Einzelzellexpression eines Gewebes kann das Zellschicksal beeinflussen. Die Bestimmung der Gewebszusammensetzungen hilft beim Verständnis genetischer Entwicklungen und Krankheiten. Gegenwärtig wird dies oft mittels Genexpressionsdaten von Einzelzellen aus Sequenzierungsexperimenten durchgeführt. Zusätzlich kann die gemeinsame Messung und Analyse weniger Zellen in kleinen Zellpools interessante zusätzliche Informationen liefern, die bei einer Einzelzellmessung, beispielsweise aufgrund niedrigen Signals, verloren gehen könnten.

In dieser Arbeit befasse ich mich mit der Frage der Modellierung der Low-Input-Genexpression von Einzelzellmessungen und solcher Small-Pool-Messungen, indem ich Erkenntnisse aus der Einzelzellanalyse übertrage. Viele Tools modellieren die Anzahl an RNA-Sequenzen mittels einer Wahrscheinlichkeitsverteilung. Diese muss so ausgewählt werden, dass sie nicht nur für die Modellschätzung, sondern auch für Interpretierbarkeit, Komplexität und biologische Plausibilität der zugrundeliegenden Annahmen geeignet ist. Um die Interpretierbarkeit zu gewährleisten, untersuche ich mechanistische Transkriptions- und Degradationsmodelle, die bestimmten Wahrscheinlichkeitsverteilungen zugrundeliegen. So genannte Bottom-up-Ansätze leiten Steady-State-Wahrscheinlichkeitsverteilungen wie zum Beispiel die Poisson- oder die Poisson-beta-Verteilung aus den zugehörigen Transkriptions-Degradations-Modellen ab (Dattani and Barahona, 2017). Indem ich dieses Verfahren umdrehe, zeige ich, wie man ein entsprechendes biologisches Transkriptions-Degradations-Modell aus einer gegebenen Wahrscheinlichkeitsverteilung ableiten kann. Dies wird mittels eines nützlichen Zusammenhangs zwischen Ornstein-Uhlenbeck-Prozessen - einer speziellen Art von stochastischen Prozessen - und stationären Wahrscheinlichkeitsverteilungen erreicht. Damit leite ich in einem Fall den zugehörigen Mechanismus her, der bei Verwendung der negativen Binomialverteilung angenommen wird. Ein realistisches mechanistisches Modell, das dieser Verteilungsannahme für die Anzahl an vorhandenen mRNA-Molekülen zugrundeliegt, ist bisher noch nicht explizit beschrieben worden, kann aber über andere Wege von ähnlichen Modellen abgeleitet werden (Berg, 1978, Paulsson et al., 2000, Raj et al., 2006). Ich zeige, dass die negative Binomialverteilung als stationäre Verteilung aus einem mechanistischen Modell stammt, das mRNA-Moleküle in Bursts erzeugt. Ich veranschauliche empirisch, dass dieses Burstingmodel und somit die negative Binomialverteilung ein geeigneter Kompromiss zwischen Rechenaufwand und biologischer Vereinfachung darstellt. Zum Vergleich werden weitere Verteilungen und mögliche zugehörige mechanistische Modelle vorgestellt, die auf unterschiedlichen Annahmen beruhen.

Darüber hinaus stelle ich ein Modell vor, mit dem man die Heterogenität einer Stichprobe bestimmen kann, indem von mehreren Zellen gemeinsam der RNA-Inhalt gemessen und analysiert wird. In der Vergangenheit wurden viele Tools entwickelt, um Bulk-Messungen mathematisch zu entfalten, wobei Informationen aus homogenen Einzelzellmessungen bzw. sortierten Bulks (Aliee and Theis, 2020) oder andere Vorkenntnisse über die Messungen benötigt werden (Gaujoux and Seoighe, 2013, Newman et al., 2015). Ich stelle hier den Stochastic Profiling Algorithmus vor, der keine Vorinformationen über die enthaltene Heterogenität benötigt und somit eine blinde Entfaltung durchführt.
Dieser Algorithmus wurde erstmals von Bajikar et al. (2014) vorgestellt und parametrisiert die Heterogenität mittels Maximum-Likelihood-Schätzung, indem er die Gesamtexpression der Zellpools entmischt. In dieser Arbeit habe ich die erste vollständige Beschreibung dieses Algorithmus zusammen mit einer Bewertung seiner Leistungsfähigkeit in breiten Simulationsstudien, unter anderem hinsichtlich der Parameterunsicherheit von Poolgrößen, erstellt. Die Einsparung von Kosten für Experimente und die Reduktion von Messfehlern kompensieren den Rechenaufwand, der beim Entmischen der Expressionsprofile durch den Stochastic Profiling Algorithmus entsteht. Zudem habe ich weitere Anwendungsmöglichkeiten in der Downstream-Analyse der entdeckten Heterogenitäten entwickelt. Somit kann man nicht nur die enthaltene Heterogenität parametrisieren sondern auch die Populationszusammensetzung bestimmter Beobachtungen vorhersagen und Unterschiede zwischen Zellpopulationen oder zwischen Stichproben genauer betrachten.
Mit dem Aufkommen der Sequenziertechnologien ist es notwendig geworden, den Algorithmus zu überarbeiten, um den neuen Anforderungen mit einem diskreten Modell gerecht zu werden. Daher habe ich eine solche neue Version unter Verwendung der negativen Binomialverteilung entwickelt, sodass der Algorithmus nun Sequenzierungsmessungen unter Berücksichtigung geeigneter Annahmen entfalten kann. Um die Unsicherheit der Modellparameter einzubeziehen, habe ich die Parameterschätzung um Bayes'sche Methoden unter Verwendung von Hamiltonian Monte Carlo erweitert. Die Berechnungseffizienz wurde mit der Stan-Implementierung des No-U-Turn-Samplers optimiert. Der Vergleich mit der bisherigen Maximum-Likelihood-Optimierung zeigt übereinstimmende Ergebnisse in kürzerer Zeit, insbesondere für komplexere Mischungen.
Abschließend zeige ich die Anwendung des diskreten Algorithmus auf echte Daten. Hier untersuche ich zwei Datensätze, die unterschiedliche Zellzahlen in ihren Messungen enthalten. Diese Daten wurden für diesen Zweck erstellt und bestehen aus einerseits homogenen embryonalen Mausstammzellen und andererseits aus Zellen von an akuter myeloischer Leukämie erkrankter Patienten, die somit vermutlich heterogene Genexpression aufweisen. Obwohl sich anhand dieser Daten weder ein reduziertes Rauschen noch eine bessere Heterogenitätserkennung bei wachsender Poolgröße feststellen lässt, ist ein ein generellen Informationsgewinn mit zunehmender Poolgröße erkennbar.

Contents

1 Introduction **1**

1.1 Overview of the Thesis . . . 1

1.2 Contributing Manuscripts . . . 3

1.3 Contributing Software . . . 5

2 Biological Background **7**

2.1 Gene Expression . . . 7

2.2 Transcriptomic Measurements . . . 10

2.2.1 DNA Microarrays Measure Relative mRNA Content . . . 10

2.2.2 Sanger Sequencing and Next Generation Sequencing Technologies Measure mRNA Counts . . . 11

2.2.3 Third Generation Sequencing . . . 13

2.2.4 Single-Cell Measurements . . . 14

2.2.5 Measurements of Small Pools of Cells . . . 15

2.2.6 Data Preprocessing . . . 16

2.3 Acute Myeloid Leukemia . . . 17

2.4 PDX Mouse Models . . . 20

3 Mathematical Background **23**

3.1 Construction of Parametric Probability Distributions . . . 24

3.1.1 Mixture of Probability Distributions . . . 24

3.1.2 Compound Probability Distributions . . . 26

3.1.3 Convolution of Distributions . . . 29

3.2 Stochastic Processes . . . 33

3.3 Parameter Inference . . . 35

3.3.1 Maximum Likelihood Estimation . . . 35

3.3.2 Bayesian Parameter Inference . . . 36

3.4 Model Selection and Goodness of Fit . . . 37

4 Modeling Single-Cell mRNA Counts **39**

4.1 Prior Work . . . 39

4.2 Inferring Distributions from Gene Expression Mechanisms . . . 43

4.2.1 Basic Model: Constant Transcription and Degradation . . . 43

4.2.2 Generalization: Time-Varying Transcription and Constant Degradation . . . 44

4.2.3 Switching Model: Gene Activation and Deactivation 46
4.3 Connecting SDEs with Steady State Distributions 49
4.3.1 Negative Binomial Distribution: Deriving an Explanatory Bursting Process 52
4.4 Further Distributions and Models 60
4.4.1 Basic-Bursting Model 61
4.4.2 Poisson-Inverse Gaussian Distribution 65
4.5 Application: Data Generation and Model Selection 69
4.5.1 Heterogeneity and Dropout 70
4.5.2 R Package scModels 71
4.5.3 Simulation Study 78
4.5.4 Application to Real-World Data 83
4.5.5 NB Distribution as Commonly Chosen Count Model 92
4.6 Discussion and Conclusion 93

5 Estimating Single-Cell Properties from Pooled Cell Data 97
5.1 Background 98
5.2 Statistical Model Underlying Stochastic Profiling 100
5.2.1 Notation 100
5.2.2 Single-Cell Models of Heterogeneous Gene Expression 100
5.2.3 Small-Pool Models of Heterogeneous Gene Expression 104
5.2.4 Likelihood Function 113
5.3 Maximum Likelihood Estimation and Model Selection 113
5.4 Simulation Studies Using the **stochprofML** Package 117
5.4.1 Simulation Study on Optimal Pool Size 117
5.4.2 Simulation Study on Impact of Parameter Values 120
5.4.3 Simulation Study on the Uncertainty of Pool Sizes 121
5.5 Bayesian Parameter Inference 124
5.5.1 Implementation of the Likelihood Function in Stan 124
5.5.2 Alternative Model: Using the Poisson-Gamma Distribution . . 126
5.5.3 stochprofML: Bayesian Inference versus ML Optimization . . . 128
5.6 Interpretation of Estimated Heterogeneity 130
5.6.1 Comparison of Inferred Populations 131
5.6.2 Prediction of sample compositions 134
5.7 Discussion and Conclusion 138

6 Application to Real-World Small Pool Data 141
6.1 Experimental Setup 141
6.2 Experiment Output and Further Data Processing 144
6.2.1 Downsampling 144
6.2.2 Batch Effects of Merged Datasets 145
6.3 Noise Reduction by Cell Pooling 146
6.4 Heterogeneity Detection by Cell Pooling 150
6.5 Discussion and Conclusion 151

7 Summary and Outlook 153

Abbreviations 159

Bibliography 161

A Probability Distributions 175
A.1 Continuous Probability Distributions 175
A.2 Discrete Probability Distributions 178

B Mathematical Identities 181

C Overview of Single-Cell Analysis Tools 183

D Master Equations 185
D.1 Master Equation of the Basic Model 185
D.2 Master Equation of the Generalized Model 188
D.3 Masterequation of the Switching Model 189

E Queueing Systems 197
E.1 Queueing System of the Basic Model 197

F Usage of stochprofML 199
F.1 Synthetic data generation 199

G Further Plots of the stochprofML Simulation Studies 203
G.1 Impact of pool sizes 203
G.2 Impact of parameter values 205

H Details on Stan fits 211

I Additional Plots and Results of SFB Data 217
I.1 Descriptive Plots and GAMLSS Results 217

1 Introduction

Curing all diseases is the hope of mankind. For some time now, the greatest possible efforts have been made to understand the human body. It is of huge importance to understand how the healthy processes work in order to deduce what goes wrong with a disease in the body. It is now known that often the smallest components of tissues - the cells - are responsible for changes by altering their function. Over time, newer and newer technical methods have been developed to look into the cells of the body and capture these changes. Especially recently, huge world wide initiatives such as the Human Cell Atlas (HCA, Regev et al., 2017) and the Human Biomolecular Atlas Program (HuBMAP Consortium., Writing Group ., Snyder et al., 2019) have joined forces to collect data in order to generate atlases of all cell types and their distinct molecular profiles in the human body. Furthermore, consortia such as the LifeTime (Rajewsky et al., 2020) initiative investigate the development on the individual cell level not only in healthy individuals but also during the progression of diseases and possible therapies. These have led to ever more and larger amounts of data. In order to be able to deal with these data, it has become more and more important to develop appropriate statistical methods. This enables us to analyze the data and to draw conclusions about biological processes through mathematical modeling, see Figure 1.1.

1.1 Overview of the Thesis

The overall goal of this thesis is to develop suitable mathematical explanations for the use of specific parametric distributions for modeling low-input mRNA sequencing counts. Although several single-cell gene expression models have been developed in the past, distributions are often used without further reference to the underlying assumptions. Especially for low-input data, i. e. when little input material or samples are available, parametric approaches are powerful. However, the parametric assumptions made thereby have a direct effect on prediction and its uncertainty. Therefore it is important to know and justify these assumptions. We approach this research question of appropriate distributions by linking suitable distributions for single-cell

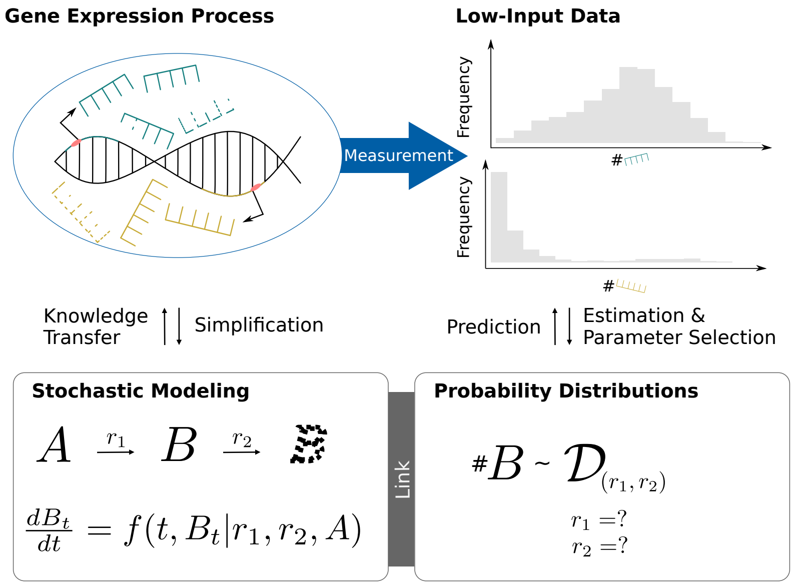

Gene Expression Process
Low-Input Data
Measurement
Frequency
Frequency
#
#
Knowledge Transfer
Simplification
Prediction
Estimation & Parameter Selection
Stochastic Modeling
$A \xrightarrow{r_1} B \xrightarrow{r_2}$
$\frac{dB_t}{dt} = f(t, B_t | r_1, r_2, A)$
Link
Probability Distributions
$\#B \sim \mathcal{D}_{(r_1, r_2)}$
$r_1 = ?$
$r_2 = ?$

Figure 1.1: Transcriptomics measures gene expression quantities in many samples e. g. single-cells. The resulting data is analyzed using statistical tools that are based on probability distributions. These should be selected with regard to the inherent assumptions on the linked stochastic model. Such models simplify the unknown gene expression generating biological process. Assessing the model fits generates further knowledge on the unknown truth of the underlying biology.

measurements with possible underlying transcription processes.

To be able to follow this close connection between mathematical and biological concepts, we present the necessary biological and mathematical background knowledge in the Chapters 2 and 3. In particular, we present details on transcriptomic measurements and principals of the stochastic processes used.

In Chapter 4, we mathematically investigate the connection between a selected probability distribution and the inherent assumptions about the underlying biological process. There, we relate statistical representations of single-cell mRNA measurements to possible mechanistic models underlying the biological transcription process. We show the linking of several models with distributions and extend those distributions to real-world circumstances such as technical errors and heterogeneity. Through their application to simulated and real-world data we can support that the often used negative binomial distribution is in fact a very suitable choice to model single-cell mRNA counts.

These findings are then transferred to joint measurements of several cells by mathematical convolution. Such cell pools might be advantageous to measure in practice. In Chapter 5 we present the statistical model of the existing stochastic profiling algorithm, called **stochprofML**, which was developed to deconvolve continuous mRNA measurements of such pools to single-cell profiles. Here we revise and extend this continuous model to varying pool sizes and include extensive simulation studies. A complete statistical description of the core model is necessary, since our extension to discrete sequencing measurements is based on it. Furthermore, we add a Bayesian extension to incorporate parameter uncertainty, and thus introduce an additional perspective. Next to the new model we also contribute to the downstream analysis when studying the inferred heterogeneity. A new test allows to compare derived distributions of different samples and a predictor specifies the pool composition of selected measurements.

An application to real-world data of the proposed discrete deconvolution model is given in Chapter 6. There we present mRNA measurements of varying small cell pools in two datasets, one containing homogeneous and the other presumably heterogeneous tissue. Although an increase in gene expression information per cell numbers can be observed, we cannot confirm the assumed noise reduction and increased chances of heterogeneity detection in these datasets.

Thereafter, we conclude in Chapter 7 our findings and offer an outlook to future possible applications and extensions of the presented methods.

1.2 Contributing Manuscripts

Some parts of this thesis have been published or submitted to peer reviewed journals and/or exist as published preprints on `https://www.biorxiv.org` or `https://arxiv.org/`. Since they were written in collaboration with co-authors, especially with colleagues from the Institute of Computational Biology, Helmholtz Zentrum München, the articles are listed in the following, along with my individual contributions that are relevant for this thesis.

- **Amrhein, L.**, Harsha, K., and Fuchs, C. (2019). A mechanistic model for the negative binomial distribution of single-cell mRNA counts. *bioRxiv 657619*

  This manuscript shows how to link suitable distributions for single-cell measurements with possible underlying transcription processes using Ornstein-Uhlenbeck processes. I conducted all mathematical derivations as described in the paper. Further I designed and conducted the simulation study and applied the models on real world datasets. I implemented the R package **scModels** with help of my student assistant Kumar Harsha, who supported me especially with C++ coding. All aspects of the project were supervised by Christiane Fuchs. The manuscript was written in cooperation with Christiane Fuchs. Apart from minor changes and the additional models that I have developed more recently, Chapter 4 and Amrhein et al. (2019) match. Some parts are contained in the mathematical background in Chapter 3 and the Appendices A-E.

- **Amrhein, L.** and Fuchs, C. (2020b). stochprofML: Stochastic Profiling Using Maximum Likelihood Estimation in R. *arXiv:2004.08809 [stat.AP].*

  In this manuscript I described the statistical model of the existing stochastic profiling algorithm, called **stochprofML**, in detail. This tool as been proposed before by Bajikar et al. (2014) and deconvolves continuous joint mRNA measurements of several cells to single-cell profiles. I extended, modified and improved the described method. With this I provided for the first time a detailed and complete description of the complete model. I improved and extended the R package **stochprofML**. I developed a new statistical test for the hypothesis that two distributions that are inferred from different sample sizes and pool sizes are the same. Additionally, I implemented a procedure to predict the population composition of small cell pools based on the inferred population parameters. I designed and conducted all simulations studies with help of my student assistants Susanne Amrhein and Xiaoling Zhang. All aspects of the project were supervised by Christiane Fuchs. The manuscript was written in cooperation with Christiane Fuchs. Apart from minor modifications, the first part of Chapter 5 and Amrhein and Fuchs (2020b) match. Since the paper discusses only the continuous models, the details of the discrete model are not described there. Some parts are contained in the mathematical background in Chapter 3 and the Appendices F and G.

- **Amrhein, L.** and Fuchs, C. (2020a). Stochastic Profiling of mRNA Counts Using HMC. *Proceedings of the 35th International Workshop on Statistical Modelling (IWSM)*

  In this manuscript I described the discrete model extension for stochastic profiling and introduce a Bayesian inference method to infer model parameters. I developed and implemented the method described in the paper with help of my student assistant Mara Santarelli, who helped especially in setting up the model in the programming language Stan. I designed and conducted the simulation study. The manuscript was written in cooperation with Christiane Fuchs. With the exception of some small adjustments and an additional simplified version of the model that was not yet included in the paper, the first part of Section 5.5.3 and Amrhein and Fuchs (2020a) are identical. Some parts are contained in the mathematical background in Chapter 3

- Tirier, S. M., Park, J., Preußer, F., **Amrhein, L.**, Gu, Z., Steiger, S., Mallm, J.-P., Krieger, T., Waschow, M., Eismann, B., Gut, M., Gut, I. G., Rippe, K., Schlesner, M., Theis, F., Fuchs, C., Ball, C. R., Glimm, H., Eils, R., and Conrad, C. (2019). Pheno-seq – linking visual features and gene expression in 3D cell culture systems. *Scientific Reports*, 9(12367).

  This manuscript describes a new sequencing technique that measures the joint gene expression of complete spheroids of colorectal cancer (CRC). During this collaboration I extended the stochastic profiling algorithm to allow different pool sizes. This method extension is part of Chapter 5. Besides implementing this new extension in our R package **stochprofML**, my part in this study was focused on its application. Therefore, I conducted the needed preprocessing of the CRC data followd by the stochastic profiling analysis. The CRC data was provided by Stephan M Tirier. All aspects of the deconvolution part of the project were supervised by Christiane Fuchs. The parts of the manuscript concerning the deconvolution of CRC pheno-seq data was written in cooperation with Christiane Fuchs, Fabian J Theis and Stephan M Tirier. All other parts of the project were performed by the remaining author team.

1.3 Contributing Software

The software and programming code generated during this thesis are published. The two R packages **scModels** and **stochprofML** are available on CRAN (`https://cran.r-project.org/`). All other relevant code can be found on the research group's GitHub repository `https://github.com/fuchslab`.

2 Biological Background

In this chapter we present important biological background information necessary to follow this thesis. This thesis is partly based on transferring existing mathematical and statistical models to biological questions. Therefore it is important to understand at least the basic biological processes one wants to model and analyze. After a basic overview of gene expression in cells we will describe the current experimental methods used to measure the abundance of the products of these processes. Especially in Chapter 6 of this thesis we will present the resulting data of such measurements. In this context, we focus our analysis on cancer, particularly on acute acute myeloid leukemia (AML), which will be introduced in the course of this chapter. The AML data presented in Chapter 6 is not directly taken from patients but stem from mouse models. Therefore, we will also introduce this specific technique employed, namely patient derived xenograft (PDX).

2.1 Gene Expression

Only about 60 years ago Crick (1958) formulated for the first time the central dogma of molecular biology, which describes the flow of the genetic information within biological systems. Although it has since then been refined (Crick, 1970), it is still valid and underlies the process of gene expression which explains how genetic information is synthesized into a gene product. The most important steps of this process in eukaryotic somatic cells are described below.

The main information in a cell is contained in its cellular deoxyribonucleic acid (DNA) inside the cell nucleus. The DNA contains the genetic information of the organism. Information carriers are the four different nucleobases, adenine (A), guanine (G), thymine (T) and cytosine (C), which encode the DNA sequence. More precisely, the DNA consists of two complementary DNA strands that contain the same information as the complementary strand is built up by the counterparts of each nucleobase of the other strand: A complements T and C complements G. Consisting of these two connected complementary strands, the DNA forms the famous double helix by coiling around its own axis (Watson and Crick, 1953). The DNA double helix is wrapped around histones and thereby forms a complex called chromatin. After additional condensation the chromatin then organizes itself into several chromosomes. This is called the genome and is generally the same in all cells of an organism.

The definition of “gene” is constantly being revised, as new findings are being gathered. Gerstein et al. (2007) recently refined that “The gene is a union of genomic sequences encoding a coherent set of potentially overlapping functional products”. In general, genes are those parts of the DNA that are transcribed into functional ribonucleic acid (RNA) molecules. These new RNA molecules are complementary copies of the DNA and built in the same way as single-stranded complementary DNA (cDNA) with the difference that the nucleobase thymine (T) is replaced by a different nucleobase uracil (U). Depending on the gene, RNA molecules can have different functions. For example, transfer RNA (tRNA) or small nuclear RNA (snRNA) directly perform tasks within a cell and therefore play a direct role for example in gene regulation or protein translation. In contrary, messenger RNA (mRNA) have a more indirect function and the mRNA sequences serve themselves as templates for proteins. During translation, the encoded amino acid sequence is built up and therefore results in a more complex structure of proteins. In most organisms, the DNA and thus the mRNA sequence codes for 20 different amino acids. Both mRNAs and proteins degrade after some time.

With this we have only roughly described the steps of gene expression and left out further intermediate steps such as for example mRNA splicing which follows directly after the transcription of mRNA. Actually, the transcribed mRNA is called precursor mRNA (pre-mRNA) and has to be converted into mature mRNA. This is done by cutting out non-coding regions of the pre-mRNA called introns and recombining the coding parts called exons, see Figure 2.1. Gene expression measurement generally refers to the quantitative measurement of either mRNAs (transcriptomics) or proteins (proteomics) or both. In this thesis, when we mention gene expression and its measurement, we usually refer to transcriptomics.

Unlike the DNA sequence, the activity of certain genes and therefore the amount of their mRNA and protein content can differ in cells. Variations of mRNA and protein content is the natural result of stochastic processes - such as mRNA transcription, protein translation and their degradation. However, different cell types perform different functions in an organism. These functions may be correlated to different gene activities which are no longer the result of purely stochastic variation. Therefore, these specific functions of the cells must have been ensured by regulatory mechanisms that are activated during cell differentiation or other environmental influences. One starting point is the modification of DNA, which is generally referred to as epigenetics. The most frequently investigated modifications are the methylome, that characterizes the methylation changes of nucleobases of the DNA and the chromatin architecture that represents possible chemical changes in the histone proteins of the chromatin. Figure 2.2 (adapted from Colomé-Tatché and Theis, 2018) shows a scheme of the aforementioned layers of gene expression. Nowadays there exist many experimental technologies to measure each of these. Measurements quantifying the proteome or transcriptome are of great use to analyze differences in gene expression between differ-

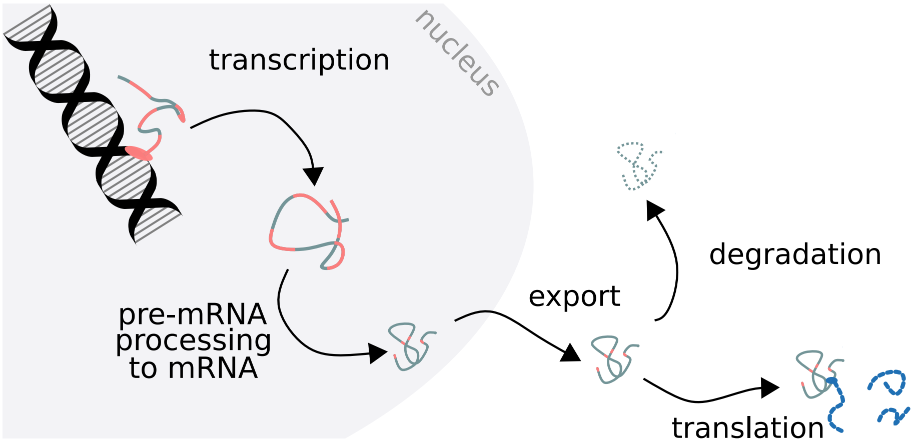


Figure 2.1: Simplified scheme of the lifecycle of an mRNA. Red lines refer to introns and untranslated regions, exons of the mRNA are depicted colored teal. The red ellipse depicts the polymerases needed for transcription that attaches to DNA. The small blue ellipse shows the ribosome that translates mRNA to proteins, depicted by a chain of small blue dashes. Freely adapted on Harries (2019) under the CC BY 4.0 license.

ent groups of cells. This includes the analysis of different cell types, the quantification of drug effects vs. place, or other environmental changes. In the next section, we will explain with more detail the methodology for the measurement of transcriptomic data. Epigenetic or genetic differences generally have an effect on the gene products, so if the cause of an effect is to be analyzed those layers might have to be measured. For example analyzing the epigenetic layer, i. e. the methylome and chromatin architecture, helps to understand differentiation processes or other environmental changes. Determining the genome and thus identifying genetic changes, helps to understand for example the role of specific genes in the development of genetic diseases. Although genetic changes such as the alterations of single nucleobases in the DNA sequence occur constantly as part of evolution and often do not affect subsequent layers, accumulation of these changes – including the deletion, duplication or shifting of whole chromosomes or parts of them – can sometimes lead to diseases such as cancer.

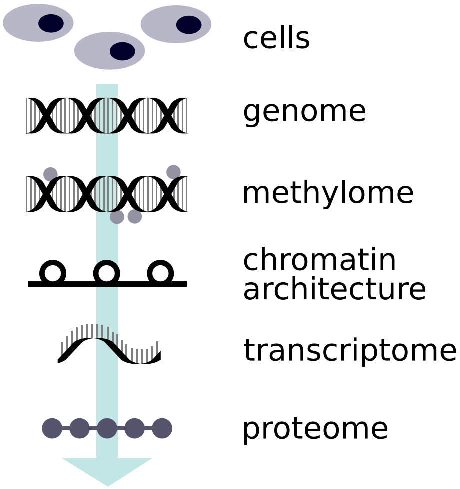


Figure 2.2: Scheme of the information flow in cells. Starting from the genetic information in the genome, the epigenetic layer defined by the methylome and the chromatin architecture governs its transcription into mRNA (transcriptome) and the subsequent translation into proteins (proteome). Nowadays, all these layers can be analyzed individually and sometimes jointly through experimental measurements. Figure adapted from Colomé-Tatché and Theis (2018).

2.2 Transcriptomic Measurements

A revolution in the measurement of gene expression has taken place in recent years, alike many other experimental methods. Most methods were originally developed for genetic measurements, e. g. the identification of DNA sequences, and then further developed and transferred to transcriptomic measurements. Determining the amount of mRNA of particular genes in a sample is important to investigate which genes are more or less active. Since the applications in this thesis are all based on transcriptomic measurements, we will present some of the most popular experimental methods and their development in recent years. In addition, we also include a small section on raw data preprocessing, i. e. how the output of the machines needs to be processed so that it can be analyzed with statistical tools.

2.2.1 DNA Microarrays Measure Relative mRNA Content

To determine the abundance of transcriptomes of many genes at once, Schena et al. (1995) showed how DNA microarrays can be used to accomplish this task. A microarray is a small chip that is covered with small wells, where each well can be marked with one specific DNA array representing one gene sequence of interest. Therefore, it is not possible to quantify an “unlimited” number of gene sequences, as the plate size (= number of wells) is the upper limit. The mRNA of a tissue sample to be analyzed needs some preparations before it can be loaded onto the microarray. Since we want to determine mRNA abundances but the wells are marked with DNA arrays, the mRNA needs to be translated into exact cDNA copies. In addition, these cDNAs are labeled with fluorescence. The material prepared in this way can then be placed onto the microarray and the affected cDNAs hybridize with the complementary DNA arrays in the wells by attaching to their counterparts. After washing the chip, all unhybridized material is washed away. Next, the fluorescence of each well is measured by scanning the chip with a laser. In this way, light intensities are obtained for each gene sequence, which can then be translated to relative frequencies compared to other measured intensities on this chip. Note that the DNA arrays on the plate need to be prepared in advance. This means that one must already know in which genes expression is to be measured and its complementary DNA sequence needs to be known. See Figure 2.3 for a schematic depiction of this process. A big disadvantage of this method is that each plate can only be compared with itself and the genes on it. There is no meaningful way to compare the intensities of a gene abundance on one plate with its intensity on a different plate. The intensities are always normalized on the plates and depend strongly on the source material, which can vary greatly from plate to plate, and on the laser used for scanning. When studying differentially expressed genes between two groups, e. g. male vs. female or sick vs. healthy, each group is labeled with its own fluorescence color, then mixed together and measured on the same plate. The mixture of the colors or tendency towards one of them shows the differences in gene expression between the two groups. Nowadays, microarray measurements are still used and are suitable, for example, to find candidate genes quickly and at low cost. Nevertheless, they are increasingly replaced by mRNA sequencing, which we will introduce in the next part. For more details on microarrays, see Schena et al. (1995), Malone and Oliver (2011) and Drăghici (2012).

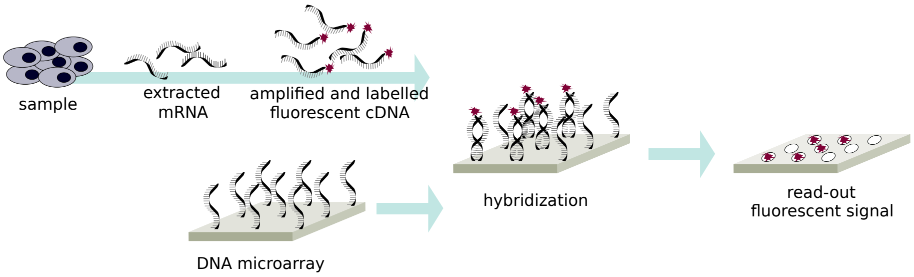


Figure 2.3: A general overview of DNA microarrays used in gene expression studies, based on Neugebauer et al. (2010). The extracted mRNA of a tissue sample needs to be reverse transcribed to cDNA. Additionally a fluorescent marker is attached to each of these cDNAs. Each well of the microarray is marked with a DNA array, containing one specific gene sequence. After loading the prepared cDNA on the microarray, affected cDNAs hybridize with the complementary DNA arrays. In order to determine relative abundance of hybridized cDNA arrays for each well, in a next step a laser measures their light intensities.

2.2.2 Sanger Sequencing and Next Generation Sequencing Technologies Measure mRNA Counts

In 1977, (Sanger et al., 1977) laid the foundation for a revolution in genomics by developing a method to read the sequence of nucleobases in DNA. In short, Sanger

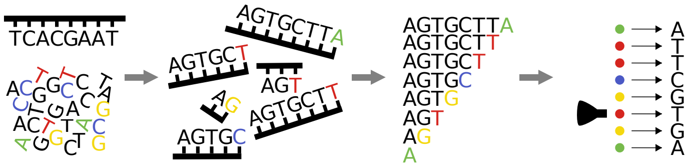


Figure 2.4: Rough illustration of the concept of Sanger sequencing. Complementary versions of the DNA sequence to analyze are produced via chain termination PCR using classical nucleotides and modified nucleotides with attached fluorescent labeling depending on the nucleotide. As soon as one of the modified nucleotides is used during PCR, the chain terminates after this position. The resulting copies of complementary DNA of different lengths are ordered by size using gel electrophoresis. A laser detects the fluorescence of the modified nucleotide, which determines the original base. In combination with the ordering the sequence can be read. Figure adapted from Muzzey et al. (2015) under the CC BY 4.0 license.

sequencing consists of three steps (see Figure 2.4). First, the complementary version of the DNA sequence of interest is generated many times using a method called chain termination polymerase chain reaction (PCR). Additionally to the classical four nucleotides of the DNA – the four deoxyribonucleotide triphosphates (dNTPs) – modified versions the so-called dideoxyribonucleotide triphosphates (ddNTPs) are mixed in. Once one of these modified versions is used during PCR, the new sequence is terminated at this point. In addition to the termination property, these ddNTPs also have a fluorescent label attached. In a second step, all replicates are sorted by size using gel electrophoresis. Then the fluorescence which is emitted by the ddNTPs and thus defines the last nucleobase of the truncated replica, is determined by a laser. Finally, the sequence of the nucleobases can be reconstructed by combining the ordering of the replicates and the fluorescence.

The term **NGS** (Goodwin et al., 2016) describes advanced high-throughput methods that make sequencing cheaper and faster and thus more useful. In contrast to Sanger sequencing, NGS methods enable the sequencing of millions of DNA pieces at once. With the transfer of sequencing methods to transcripts – now called RNA-seq – , a big step was taken to determine the amount of RNA molecules in a sample (Wang et al., 2009). The main challenge is to create the so-called library, which contains the synthetical cDNA, that corresponds to the RNAs of interest, but which are more stable and can be sequenced with the usual available NGS methods. In contrast to DNA sequencing, where one wants to determine the exact DNA sequence, the main interest in RNA sequencing is to quantify mRNAs of the same gene. Using these quantities one wants to determine highly expressed genes or to analyze different gene expression between individuals or treatments. As there exist many different protocols for library preparation and different NGS methods, the steps taken might differ. In most protocols RNA is amplified several times before cDNA conversion so it is less likely to loose some RNAs (including its replicates) during the process. To speed up sequencing, often the mRNAs from one sample get attached sample specific barcodes, so that sequencing of several samples can be performed together and afterwards the reads can be separated by these barcodes again.

2.2.3 Third Generation Sequencing

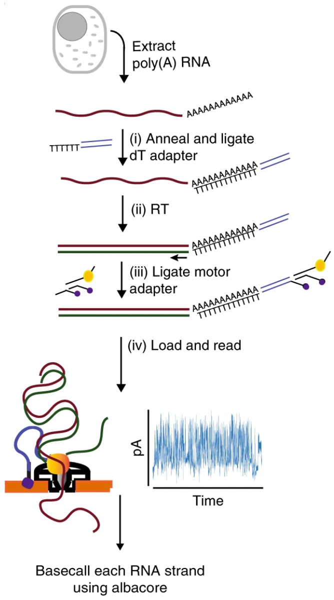


Figure 2.5: Direct sequencing of poly(A) RNA strings. As the method was originally developed of double stranded DNA, a dT adapter is attached to the poly(A) tale of the extracted mRNA and with reverse transcription a cDNA strand is transcribed. A motor adapter is attached which navigates to the nanopore and ensures that only the original mRNA strand gets passed through it. The current is monitored and the mRNA sequence of interest can be determined. Figure taken from Workman et al. (2019).

Sequencing technology is still developing. There are constantly new protocols being published that allow to process more samples, increase the sequence depth and/or decrease processing times and prices. These new techniques are called *third-generation sequencing* methods. Nanopore sequencing (Goodwin et al., 2016) is such an example. Already described many years ago, newer technologies allow nowadays to efficiently use this method. Initially it was constructed for the characterization of DNA strands. The DNA string of interest is directly sequenced and no transformation to cDNA is necessary. Also amplification is not needed. Roughly speaking the idea is the following: A motor adapter is attached to the double helix and directs to the nanopore. In fact the nanopore is a lipid membrane with many small (nanometer size) holes in it, the pores. As soon as the adapter attaches to such a pore, the DNA strands are up and one is directed through the pore. When the strand passes through the pore an attached current is recorded as it changes for different bases of the passing DNA. Afterwards when analyzing the current shifts one can reconstruct the DNA sequence. Nowadays this technology is transferred to directly sequence RNA reads. As Figure 2.5 shows, one reverse transcription step that generates cDNA needs to be performed, but this is only for further adapter ligation, as it is the original RNA that passes through the nanopore and hence gets directly sequenced. In contrast to many NGS methods, long reads are no problem, as the sequences to be determined are not fragmented. But also nanopore measurements are challenging, as it is hard to determine each base separately. This can be improved by either slowing down the string passing through the pore or by improving its sensitivity. A nanopore sequencing machine is very small and portable. As it is relatively cheap and easy to use, this technology has the potential for fast analysis of genomes and transcriptomes.

2.2.4 Single-Cell Measurements

In general, the methods described above have been developed for bulk samples, i. e. tissue samples containing thousands to millions of cells. These bulk measurements are sufficient for many applications, such as genotyping of individuals or determining changes in gene expression before and after treatment. However, gene expression is not only heterogeneous between individuals and cell types, but also within cells of the same type. Single-cell data appears to be better suited to fully identify heterogeneity. Therefore, there has been great interest in developing methods to sequence the information contained in each individual cell. In general, single-cell methods have to overcome two main difficulties during library preparation: selection for single cells and dealing with the small amount of input material. This has been solved for single-cell microarrays (Kurimoto, 2006, Tietjen et al., 2003), as well as for **scRNA-seq** (Kolodziejczyk et al., 2015, Sandberg, 2014). Since we are only dealing with sequencing data in this thesis, we will only focus on scRNA-seq technologies in the following. The basic procedure that is common to all scRNA-seq experiments is to first capture and lyse a single cell. After the reverse transcription step when mRNA molecules are selected by poly(T) priming, the necessary cDNA is obtained, which is then amplified by PCR or in vitro transcription. This amplified cDNA forms the basis for the preparation of the sequencing library (Kolodziejczyk et al., 2015). Each protocol approaches these steps slightly differently (Ziegenhain et al., 2017). Depending on the goal of the measurements a different library might be appropriate. In Chapter 6 we will present data generated by our collaboration partners using their plate-based improved single-cell RNA barcoding and sequencing (SCRB-seq) protocol (Soumillon et al., 2014), called molecular crowding SCRB-seq (mcSCRB-seq) protocol (Bagnoli et al., 2018). As shown in Figure 2.6, single-cells are isolated in multi-well plates by fluorescence-activated cell sorting (FACS). The wells contain some material so that the mRNA is extracted from the cells and reverse transcription is performed. In that step the cDNA that is complementary to the targeted mRNA is generated. Barcodes for the wells and the UMIs are also attached. Here the authors improved SCRB-seq by adding polyethylene glycol (PEG8000) to the content of the lysis puffer to make ligation more efficient. Next, all wells are pooled and PCR amplification is performed, which was improved by using Terra polymerases. Afterwards the prepared library is sequenced as usual in a sequencing machine.

Plate-based technologies existed prior to single-cell sequencing, but new technologies tailored to single-cell measurements have also been developed. One of the key contributions is the development of droplet-based technologies such as the 10x Chromium (Zheng et al., 2017) where tens of thousands of cells can be measured in one single experiment. There each individual cell is captured in a microfluidic droplet in a fast, automated manner. However, often errors occur and droplets do

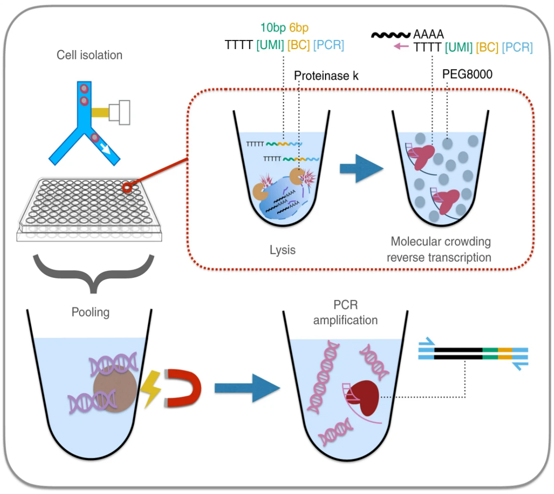


Figure 2.6: Schematic overview of the mcSCRB-seq protocol workflow. Cells are sorted via FACS into the plates which contains the lysis buffer and reverse transcription using molecular crowding is started. The barcoded material of all wells is pooled and PCR amplification is performed. Figure taken from Bagnoli et al. (2018) under the CC BY 4.0 license.

not contain any cell since the flow of cells is slow to avoid doublets, triplets,... i. e. more than one cell per droplet. With this sample sizes increased and thus data grew bigger and bigger (Angerer et al., 2017) but also cheaper.

Another major contribution in the field of scRNA-Seq was the introduction of unique molecular identifiers (UMIs), see Islam et al. (2014) and Ziegenhain et al. (2018). In addition to the barcode that identifies each sample, a unique identifier is attached to each original mRNA molecule before amplification. Therefore, all mRNA reads originating from the same mRNA molecule can be collapsed onto their UMI and thus not only the number of reads but also the number of UMIs of each gene of each sample can be counted and analyzed (see Figure 2.7). The usage of UMIs reduces the bias introduced by the amplification step, where some molecules may be amplified more often than others.

2.2.5 Measurements of Small Pools of Cells

Single-cell data is more cost-intensive and prone to technical noise than bulk measurements where millions of cells are sequenced together. There, in contrast to single-cell measurements, heterogeneous gene expression is averaged. A suitable trade-off between the bulk and single-cell approach is the joint measurement of small pools of cells. Janes et al. (2010) proposed to use single-cell techniques to measure a random selection of a specific number – e. g. 10 cells – in one sample together at this time using microarray measurements. Singh et al. (2019) further developed this idea and built a protocol called 10cRNA-seq which extends scRNA-seq to jointly sequence 10 micro-dissected cells.

Joint measurement of cells adds information to single-cell sequencing data. For example, Tirier et al. (2019) measure and image clonal tumor spheroids grown from

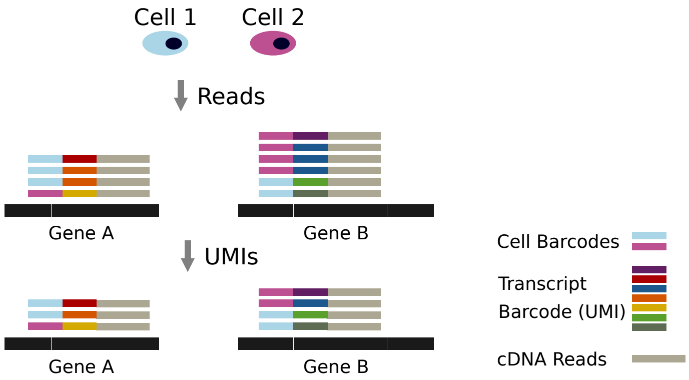


Figure 2.7: cDNA reads (gray) with attached cell barcodes (light blue and magenta) and UMIs (other colors) are assigned to two gene sequences. Collapsing reads to UMIs and correctly assigning them to single cells requires to control for sequencing errors in barcodes and UMIs. Figure adapted from Ziegenhain et al. (2018) under the CC BY-NC 4.0 license.

single-cells using their technique called pheno-seq. The size of these spheroids span a wide range from a few cells up to over 200. Through their analysis, morphological features could be identified that were lost in single-cell measurements.
Another growing field that measures several cells simultaneously is spatial transcriptomics (Asp et al., 2020). Since it is not yet possible to mark the location of the same single-cells that will be sequenced later, slides are extracted from the tissues, imaged and then localized spots containing multiple cells are sequenced together. Experimental techniques to do this are for example tomo-seq (Junker et al., 2014) or 10x visium (10x Genomics $^{\text{TM}}$, 2018) which was originally developed by Ståhl et al. (2016).
In Chapter 5 we will present a methodological approach for the analysis of small pool measurements.

2.2.6 Data Preprocessing

In the previous sections on sequencing, we stopped at the point where the library was fed into a sequencing machine and just assumed any result. In reality, this step is much more complicated. The sequencer output contains several files with lots of information. The mapping of the sequenced reads to a reference genome has not yet been done, so a look at the output files does not give any information about specific gene expressions. Parekh et al. (2018) developed a pipeline called zUMIs to process such raw RNA-seq data from any sequencer. This is the pipeline we used for data preprocessing for the data shown in Chapter 6. As shown in Figure 2.8, first, the file containing the sequenced barcodes of the wells and UMIs is combined with the file containing the sequenced cDNA reads. Quality controls are added to filter out bad reads, i. e. cases where the sequencer has most likely made a mistake. Then, cDNA sequences are mapped to the reference genome using the STAR (Spliced Transcripts Alignment to a Reference) software (Dobin et al., 2013). Many reads cannot be mapped, either because they have errors or because they were not mRNA.

In the counting step, the mapped reads are further divided into exonic and intronic or both – depending on which part of the gene their read spans – and are collapsed to their UMI numbers. The zUMIs output then provides read and UMI count tables for these exons, intros, and intron.exon numbers. These are the data files that are then analyzed further and their quality is summarized in some descriptive plots.

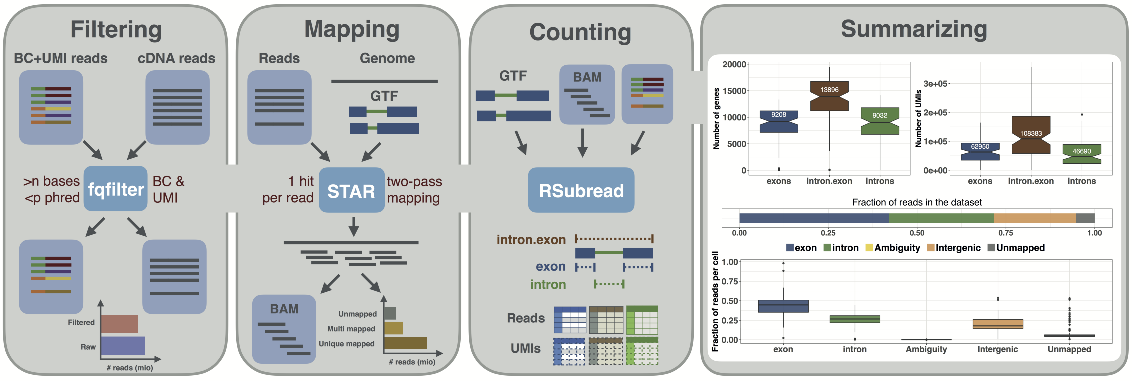


Figure 2.8: Schematic representation of the zUMIs pipeline. First the two input data files containing the reads are merged and filtered for bad reads. STAR is used to map the resulting merged reads to the reference genome. The mapped reads are then distributed to the original samples and molecules through their well and UMI-barcodes and count matrices are generated. Finally zUMIs offers first descriptive summary plots. Figure taken from Parekh et al. (2018) under the CC BY 4.0 license.

Of course there are other raw data processing pipelines that perform in general very similar steps. For example Cell Ranger (Zheng et al., 2017) was tailored for raw data preprocessing of single-cell 10x chromium data. After this as soon as read or UMI count matrices are generated, further data preprocessing such as normalization or batch correction starts and moves on to the actual data analysis. For this other pipelines such as Seurat (Butler et al., 2018) or Scanpy (Wolf et al., 2018) have been developed. With this nowadays data processing is pretty standardized. For more information we refer to a review on best practices from Luecken and Theis (2019).

2.3 Acute Myeloid Leukemia

Leukemia was recognized by John Hughes Bennett in 1845, when he recognized leukemia as a clinical disease entity by realizing that people died from suppuration of the blood (Bennett, 1845). At the same time Rudolf Virchov established in parallel the name ‘Leukhemia‘ – Greek word of white blood – for the disease as he realized the existence of more white blood cells in his patients as usual. Leukemia is a disease that originates in the bone marrow, where blood cells are produced (Geary, 2000). Blasts are immature-looking cells that also occur in normal bone marrow, but with a frequency lower than 2 % as early blood cell precursors. Actually, leukemia describes a whole group of diseases of the bone marrow with varying degrees of severity and symptoms. Therefore two classes are distinguished depending on the origin of the disease, see Figure 2.9 for healthy haematopoiesis. The lymphoid leukemia originates

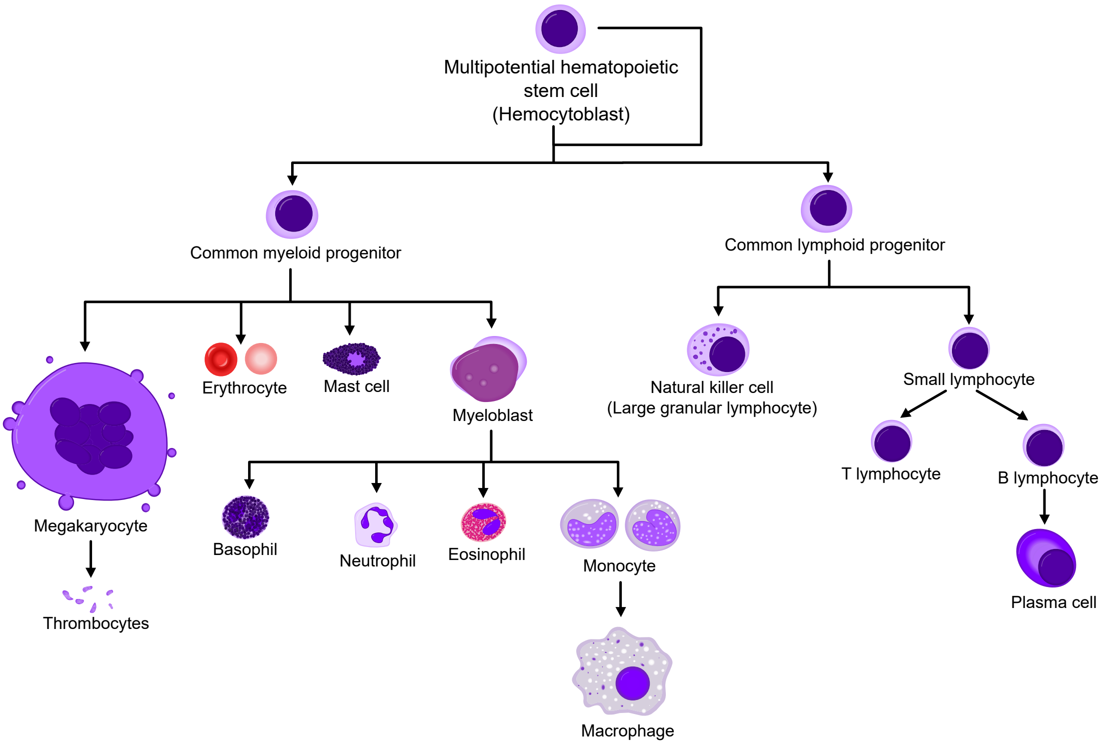

Multipotential hematopoietic stem cell (Hemocytoblast)
Common myeloid progenitor
Common lymphoid progenitor
Megakaryocyte
Thrombocytes
Erythrocyte
Mast cell
Myeloblast
Basophil
Neutrophil
Eosinophil
Monocyte
Macrophage
Natural killer cell (Large granular lymphocyte)
Small lymphocyte
T lymphocyte
B lymphocyte
Plasma cell

Figure 2.9: The healthy development from haematopoietic stem cells to mature blood cells. The lymphoid and the myeloid progenitors divide differentiation into the two developmental groups. Figure of the simplified hematopoiesis shared by A. Rad and M. Häggström under the CC BY-SA 3.0 license.

in the development of lymphoblasts, i. e. immature lymphocytes. If the myeloid line, i. e. the progenitors of erythrozytes, monocytes and megakaryocytes is affected, then we are dealing with a myeloid leukemia. Depending on the course of the disease, a distinction is made between chronic and acute leukemias. Chronic leukemias develop slowly and often do not show any symptoms in the early stages. The produced blasts are relatively mature but still abnormal white blood cells. In contrast, acute leukemias are fast developing diseases with severe disease progression and high mortality. They are characterized by a rapid crowding of blasts in the bone marrow which leaves only a very small space for healthy blood cells. Most symptoms are not directly caused by malignant blasts but by a lack of healthy blood cells. Missing white blood cells cause infections and fever, missing red blood cells are associated with a feeling of fatigue and weakness, while bleeding is caused by missing platelets. In general if blasts are increased to more than 20% an acute leukemia is present. Taken together, this leaves four main categories of leukemias: acute lymphoblastic leukemia (ALL), chronic lymphocytic leukemia (CLL), AML and chronic myeloid leukemia (CML). These can be subdivided into many more subtypes. In general – as they develop over time – chronic leukemias occur more often in elderly patients. ALL is the most common cancer in children and affects most frequently small children. In the last decade ALL is seen as the first neoplastic cancer that could be cured in children, meaning that the survival rate nowadays is over 90% (PDQ ® Pediatric Treatment Editorial Board, 2020). In contrast, AML is the most frequent acute leukemia in adults with a steep increase in age which will be a bigger issue in an aging society, see Figure 2.10.

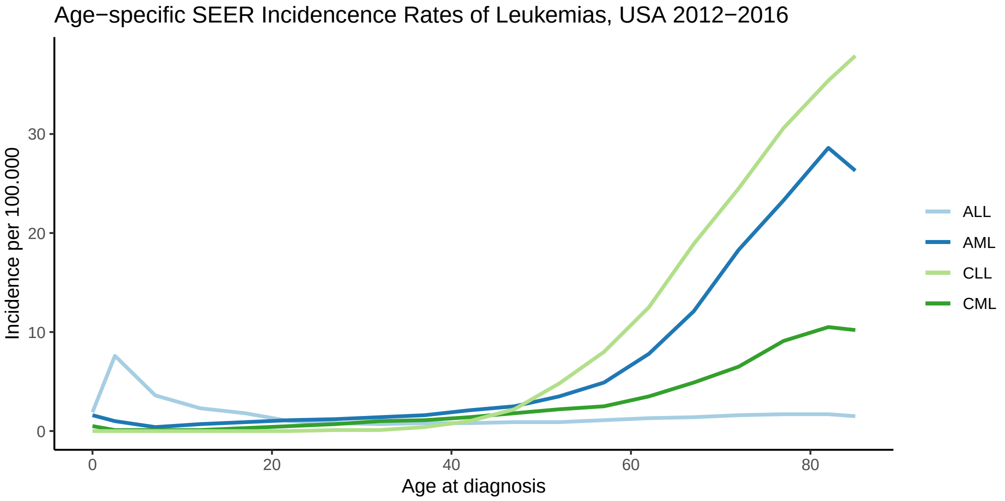


Figure 2.10: Age-specific incidence rates for the four main leukemia classes: Acute lymphoblastic leukemia (ALL), chronic lymphocytic leukemia (CLL), acute myeloid leukemia (AML) and chronic myeloid leukemia (CML). Updated data from Howlader et al. (2016).

Since in Chapter 6 we will present AML data, in the following we will focus on this type of leukemia. An AML occurs when a progenitor cell in the bone marrow accumulates many different changes in its DNA, proliferates and takes over a part in the bone marrow. These cells are then called a clone. AML is a disease where genetic factors play a role and thus some people are more likely to develop AML. But environmental factors also play a major role. Since risk factors include radiation exposure, secondary therapy-related AML can occur after myelodysplastic disease such as myelodysplastic syndrome (MDS) or after chemotherapy as well as radiotherapy for other types of cancer (PDQ ® Adult Treatment Editorial Board, 2020). For all these reasons, AML is highly heterogeneous and each AML is unique in its features. The WHO has classified many AML subtypes, but in general two AMLs from two randomly selected patients are very different and do not show the same genetic changes. Such a leukemic cell population is called a clone. Since 2008, the WHO classifies AML to be present when more than 20% of myeloid blasts can be found in blood or bone marrow (PDQ ® Adult Treatment Editorial Board, 2020). To prove that the blasts seen are myeloid blasts, one uses chemical stains (peroxidase stains) and flow cytometry. In AML there is always a founding clone that started the AML. This founding clone proliferates and occasionally accumulates new mutations, and forms so-called subclones. Hence, inside an AML many parallel subclones can exist. To cure AML a therapy has to target all types of subclones. Surviving clones can cause a relapse of the disease (as shown in Figure 2.11).

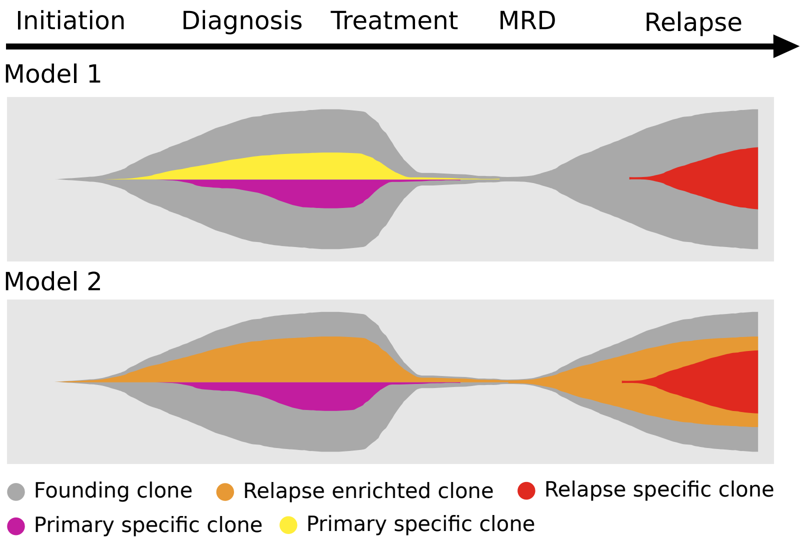


Figure 2.11: Scratch of clonal evolution over time in AML. Two major patterns are depicted that are observed most frequently. The first model shows the dominant founding clone evolving to give rise to the relapse clone while clones present at diagnosis vanish. In contrary, in model 2 a minor clone that originally evolved from the founding clone survives the treatment, stays undetected as minimal residual clone during the minimal residual disease (MRD) period and expands at relapse to finally give rise to the relapse specific clone. Figure adapted from Ding et al. (2012) under the BY-NC-SA 3.0 license.

2.4 PDX Mouse Models

The AML data treated in Chapter 6 was generated using PDX mouse models. Therefore we will shortly give an overview on cancer mouse models. In the past two different murine models have been developed: genetically engineered mouse model (GEMM) and xenograft models (Richmond and Su, 2008). Both are used for different purposes of a study. In GEMM specific genes of interest are altered so that they are mutated, deleted or over-/underexpressed. Typically these are genes that are assumed to be important for example in tumor development. Afterwards the mice are studied for tumor development over some time. In contrast, in human xenograft models, human cancer cells are transplanted into immunosuppressed mice. These are mice that lack a working complete immune system and thus do not reject the implanted human cells. As these mice carry for example cancer cells of a specific patient, one can study their response to different therapies in vivo. Xenograft models can be subdivided into two categories depending on whether they are cell line derived or patient derived xenograft models.

In detail the AML cells that we will analyze in Chapter 6 were generated as follows: human leukemia cells are transplanted into immunosuppressed mice, which in this case serve as a kind of “incubator”. The human cells can settle in the bone marrow niche and start to proliferate. After several weeks to months, 1 million cells have become 50 million cells, which we isolate from the bone marrow of the mice and use for analysis - or transplant into the next mouse for further amplification. More information can be found in Vick et al. (2015). During the sequencing process it was then important to separate the human cells from the murine cells. Therefore, the murine cells were depleted by “Mouse Cell Depletion MACS” (Miltenyi) and the human cells were sorted by FACS ARIA to obtain the appropriate cell count. More information on these purification steps are explained in Ebinger et al. (2020) and Ebinger et al. (2016).

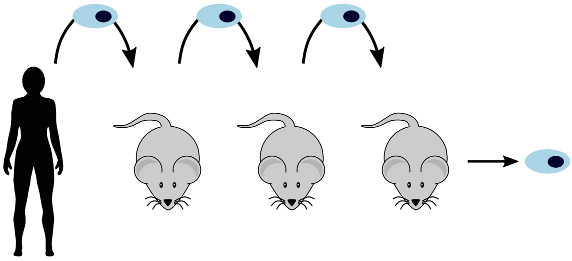


Figure 2.12: Scheme of the process of generating transgenic PDX (t-PDX) AML cells. PDX cells were transduced after first or second retransplantation cycle. Figure adapted from Vick et al. (2015) under the CC BY 4.0 license.

3 Mathematical Background

In this thesis we apply different concepts of mathematical modeling to biological questions and medical data. Data modeling can be approached from different perspectives: The procedure used in stochastics and probability theory is generally driven by the data generating process. After proposing a mechanistic and/or stochastic process that describes the individual components of the mechanism, one is interested in what the data generated by this process would look like. These can be simulated and then be compared with real data. Note that although in general a mechanistic process can also be deterministic, in this thesis we always consider it to be stochastic. In contrast to many tangible mechanistic models as known and widely used in physics, it is not as easy to build and verify such a model for biological processes.
Therefore, a different perspective is often chosen, rooted in traditional statistics, where the focus is on the given data and the question of which model best fits the data. This model does not necessarily describe the underlying mechanistic process but can also be a parametric distribution. We include both perspectives in this thesis. In this chapter we give important background information that is needed to follow the derivations of the mathematical models in the course of this thesis.

First, we introduce how new parametric distributions can be constructed using known distributions (see Appendix A). Some of these distributions will appear in Chapter 4 where we link distributional assumptions of mRNA counts to possible underlying transcriptional processes. Deeper knowledge, especially of the convolution of distributions is important to follow Chapter 5 where the described stochastic profiling algorithm deconvolves data measurements using parametric distributions.
Afterwards, we give a short introduction to the Ornstein-Uhlenbeck processes, a special form of stochastic differential equations. These are needed in Chapter 4 where we link parametric steady state distributions and mechanistic transcription processes.
Parameter inference is required to fit parametric distributions or models to given data measurements. This model parameter estimation is a large scientific area where we will only roughly sketch the main components of frequentist and Bayesian inference.
In general, different models can be fitted to some given data set. Therefore, in the last section of this chapter we will show how to select the best model. We also show

how to reject a forced model fit where the data is not modeled well enough by a goodness-of-fit test. Model fitting and selection is required in all parts of this thesis.

3.1 Construction of Parametric Probability Distributions

A major part of this thesis is focused on the appropriate statistical modeling of data: To be able to draw some conclusions about a given data set, the right software for its analysis must be selected. This selection requires knowledge of the underlying mathematical assumptions of the software, which the data must fulfill. For example, if a given data set contains continuous values, only software with underlying model assumptions applicable to continuous data should be selected. Conversely, if an algorithm is constructed to analyze certain data, the nature of that dataset must be respected. All models shown in this thesis are so-called parametric models, i. e. they are based on parametric probability distributions. Since parametric models generally give more power to the result of the analysis, they should be preferred over non-parametric models whenever possible.
Probability distributions and other mathematical terms are often not uniquely defined in the literature. Therefore all standard continuous and discrete distributions and their parameterization used in this thesis are listed in Appendix A. However, in this section we will present some more complicated parametric probability distributions. Using the univariate distributions listed in Appendix A, we introduce further distributions that are constructed by combining several of them by mixing, convolving or compounding. References include Dormann (2013), the NIST library (Olver et al., 2019), Karlis and Xekalaki (2005), and Graham et al. (2017).

3.1.1 Mixture of Probability Distributions

When talking about heterogeneity, e.g. when a sample contains gene expression measurements of different cell types, it is common to use a separate probability distribution for each underlying population. This leads directly to mixture distributions. Since typically continuous and discrete distributions are not mixed in one model, f describes in the following either a probability density function (PDF) or a probability mass function (PMF).

Definition 3.1 (Mixture Distribution) *X follows a T-fold mixture distribution (Tmix) if*

$$X \overset{iid}{\sim} \begin{cases} \mathcal{D}_1 & \textit{with probability } p_1 \\ \mathcal{D}_2 & \textit{with probability } p_2 \\ \vdots & \\ \mathcal{D}_T & \textit{with probability } p_T, \end{cases}$$

where $\mathcal{D}_1, \dots \mathcal{D}_T$ are T distributions with PDFs or PMFs $f_1, \dots, f_T$ with mixing weights $\boldsymbol{p} = (p_1, \dots, p_T)$, where $p_T = 1 - p_1 - \dots - p_{T-1}$. The resulting PDF or PMF is given by

$$f_{Tmix}(x \mid \boldsymbol{\theta}, \boldsymbol{p}) = p_1 f_1(x|\theta_1) + \ldots + p_h f_h(x|\theta_h) + \ldots + \left(1 - \sum_{h=1}^{T-1} p_h\right) f_T(x|\theta_T),$$

where $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_T\}$ are the (not necessarily disjoint) distribution parameters of the T distributions.
A random variable $X \sim \mathrm{Tmix}(\boldsymbol{\theta}, \boldsymbol{p})$ has expectation and variance

$$\begin{aligned}
\mathrm{E}_{\mathrm{Tmix}}[X] &= \sum_{h=1}^{T} p_h \mathrm{E}_{f_h}[X] \\
\mathrm{Var}_{\mathrm{Tmix}}[X] &= \sum_{h=1}^{T} p_h (\mathrm{Var}_{f_h}[X] + \mathrm{E}_{f_h}[X]^2 - \mathrm{E}_{\mathrm{Tmix}}[X]^2)
\end{aligned}$$

As a general rule, we assume $f_i(x|\theta_i) \neq f_j(x|\theta_j)$ for all i, j. This includes the possibility that distributions are the same, i. e. $f_i = f_j$ but come with different distribution parameter $\theta_i \neq \theta_j$ for $i \neq j$.

Example 3.1 (Mixture of Two Populations) *If a sample consists of two populations and the measurements of each of these populations are modeled by one single distribution $\mathcal{D}_1$ and $\mathcal{D}_2$, with PDFs or PMFs given by f_1 and f_2, which are parameterized via θ_1 and θ_2. Then a measurement of this mixture is distributed according to $pf_1(\theta_1) + (1-p)f_2(\theta_2)$ with $p \in [0, 1]$. With a probability of p the distribution of this measurement is thus $\mathcal{D}_1$, otherwise $\mathcal{D}_2$. The corresponding mixture PDF or PMF is given by*

$$f_{2mix}(x|\boldsymbol{\theta} = \{\theta_1, \theta_2\}, \boldsymbol{p} = (p, (1-p))) = p\, f_1(x|\theta_1) + (1-p) f_2(x|\theta_2).$$

One special case of mixture distributions are zero-inflated distributions. If there are many (more than expected) zeros in the data, an indicator function with point mass at zero can be introduced to model the first mixture. If the data to be analyzed is continuous but contains zeros, a zero-inflated distribution should be used. This is one of the typical cases where a discrete distribution (the zero part) is mixed with a continuous distribution. The corresponding PDF or PMF reads

$$f_{\text{zi-Tmix}}(x|\boldsymbol{\theta}, \boldsymbol{p}) = p_1 \mathbb{1}_{\{0\}}(x) + p_2\, f_1(x|\theta_1) + \ldots + \left(1 - \sum_{h=1}^{T-1} p_h\right) f_{T-1}(x|\theta_{T-1}).$$

Example 3.2 (Univariate Distribution with Zero-Inflation) *If there is only one population but for some technical reason there are more zeros than expected in the data, the PDF or PMF corresponding to the zero-inflated distribution is given by*

$$f_{zi}(x|\theta, p) = p\, \mathbb{1}_{\{0\}}(x) + (1-p) f(x|\theta).$$

In Chapter 5 we will use continuous mixtures consisting of lognormal (LN) and/or exponential (EXP) distributions and discrete mixtures using negative binomial (NB) distributions. Although, in general, all distributions contained in one mixture can be different, often only one type of distribution (with different parameters) is used for the mixture.

3.1.2 Compound Probability Distributions

Another way to create a new probability distribution is compounding. In general, both continuous and discrete distributions can be used, but care must be taken to ensure appropriate choices.

Definition 3.2 (Compound Distribution) *A compound distribution (Comp) is obtained when a random variable X follows some parameterized distribution $\mathcal{D}_1$ with PDF/PMF $f_1(x|\lambda)$ with unknown parameter λ. This distribution parameter λ itself follows some parameterized distribution $\mathcal{D}_2$ with PDF/PMF $f_2(\lambda|\theta)$, which in turn is parameterized by some parameter θ. The resulting PDF or PMF is given by*

$$f_{\text{Comp}}(x|\theta) = \int_{-\infty}^{\infty} f_1(x|\lambda) f_2(\lambda|\theta) \mathrm{d}\lambda.$$

Note that if λ is a discrete parameter, $\mathcal{D}_2$ must be a discrete distribution and f_2 is a PMF. In this case the integral changes to a sum and the new PDF or PMF f_{Comp} is a special case of a mixture distribution (see Definition 3.1) where T corresponds to all the countable values that $\mathcal{D}_2$ can take.

In this thesis we will only use compound Poisson distributions, i. e. $f_1 \equiv f_{\text{Pois}}$.

Definition 3.3 (Compound Poisson Distribution) *A compound Poisson distribution is a Poisson distribution with intensity parameter λ, which is not a constant but itself follows a distribution with PMF f, parameterized by θ. Hence, the PMF of X is given by*

$$f_{\text{Comp-Pois}}(x|\theta) = \int_{0}^{\infty} \frac{e^{-\lambda}\lambda^x}{x!} f(\lambda|\theta) \mathrm{d}\lambda \qquad \textit{for } x \in \mathbb{N}_0.$$

A random variable $X \sim \text{Comp-Pois}(\theta)$ has expectation and variance

$$\mathrm{E}_{\text{Comp-Pois}(\theta)}[X] = \mathrm{E}_{f_\theta}[\lambda] \qquad \textit{and} \qquad \mathrm{Var}_{\text{Comp-Pois}(\theta)}[X] = \mathrm{E}_{f_\theta}[\lambda] + \mathrm{Var}_{f_\theta}[\lambda]$$

Note that these distributions are sometimes called mixed distributions or in the latter case mixed Poisson distributions (see, e.g. Karlis and Xekalaki, 2005). To avoid confusion with the mixture distributions in the previous Chapter 3.1.1, we call them compound distributions in this thesis or just combine the names of the compound distributions. Next, we will list the specific compound Poisson distributions used in this thesis and derive their probability functions and other properties.

Example 3.3 (Poisson-Gamma (PG) Distribution) *The most popular example of an compound Poisson distribution is the (compound) Poisson-gamma (PG) distribution, i. e. the intensity parameter of the Poisson distribution itself follows a gamma distribution (Definition A.3). Since the resulting PG distribution is equivalent to the NB distribution (Definition A.8), they can be transformed into each other by re-parameterization. To show this, we start with the construction of a PG distribution. Let* $\alpha, \beta > 0$ *and* $x \in \mathbb{N}_0$*. Then, according to Definitions 3.3 and A.3,*

$$\begin{aligned} f_{PG}(x|\alpha,\beta) &= \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!} f(\lambda|\alpha,\beta)\mathrm{d}\lambda = \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!}\frac{\beta^\alpha\lambda^{\alpha-1}e^{-\beta\lambda}}{\Gamma(\alpha)}\mathrm{d}\lambda \\ &= \frac{1}{x!}\frac{\beta^\alpha}{\Gamma(\alpha)}\int_0^\infty e^{-\lambda(1+\beta)}\lambda^{x+\alpha-1}\mathrm{d}\lambda. \end{aligned}$$

Substitution with $u = \lambda\,(1+\beta)$ *and* $\frac{\mathrm{d}\lambda}{\mathrm{du}} = \frac{1}{1+\beta}$ *and use of* $\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}dt$ *for* $k > 0$ *leads to*

$$\begin{aligned} f_{PG}(x|\alpha,\beta) &= \frac{\beta^\alpha}{x!\Gamma(\alpha)}\int_0^\infty e^{-u}\left(\frac{u}{1+\beta}\right)^{x+\alpha-1}\frac{1}{1+\beta}\mathrm{du} = \frac{\beta^\alpha}{x!\Gamma(\alpha)}\frac{1}{(1+\beta)^{x+\alpha}}\Gamma(x+\alpha) \\ &= \frac{\Gamma(x+\alpha)\beta^\alpha}{x!\Gamma(\alpha)(\beta+1)^{x+\alpha}} = \binom{x+\alpha-1}{x}\left(\frac{1}{\beta+1}\right)^x\left(\frac{\beta}{\beta+1}\right)^\alpha \\ &= f_{\mathrm{NB}}\left(x\middle|\alpha,\frac{\beta}{\beta+1}\right). \end{aligned}$$

This is the PMF of the NB distribution (Definition A.8). With regard to the NB parameters, the PG parameters are thus given by

$$f_{\mathrm{NB}}\,(x|r,p) = f_{PG}\left(x\middle|r,\frac{p}{1-p}\right) \quad \textit{for } r \in \mathbb{R}^+ \textit{ and } p \in (0,1).$$

Another compound Poisson distribution that we will use in this thesis is the Poisson-beta distribution.

Example 3.4 (Poisson-Beta (PB) Distribution) *The (compound) Poisson-beta (PB) distribution, is a Poisson distribution where the intensity parameter follows a beta distribution (Definition A.5). Note that here we use the generalized form of the beta distribution with four parameters. The resulting PB distribution is not as famous as the previous PG distribution, but appears in some literature, often without any name (Raj et al., 2006). The resulting PMF can be calculated as shown in the following:*

$$\begin{aligned} f_{\mathrm{PB}}(x|\alpha,\beta,a,c) &= \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!}f(\lambda|\alpha,\beta,a,c)\mathrm{d}\lambda = \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!}\frac{(\lambda-a)^{\alpha-1}(c-\lambda)^{\beta-1}}{(c-a)^{\alpha+\beta-1}B(\alpha,\beta)}\mathrm{d}\lambda \\ &= \int_a^c \frac{e^{-\lambda}\lambda^x}{x!}\frac{(\lambda-a)^{\alpha-1}(c-\lambda)^{\beta-1}}{(c-a)^{\alpha+\beta-1}B(\alpha,\beta)}\mathrm{d}\lambda. \end{aligned}$$

Often $a = 0$*. With this we can proceed and simplify*

$$f_{\mathrm{PB}}(x|\alpha,\beta,0,c) = \int_0^c \frac{e^{-\lambda}\lambda^x}{x!}\frac{(\lambda)^{\alpha-1}(c-\lambda)^{\beta-1}}{(c)^{\alpha+\beta-1}B(\alpha,\beta)}\mathrm{d}\lambda = \int_0^c \frac{e^{-\lambda}\lambda^x}{x!c}\frac{\left(\frac{\lambda}{c}\right)^{\alpha-1}\left(\frac{c-\lambda}{c}\right)^{\beta-1}}{B(\alpha,\beta)}\mathrm{d}\lambda$$

Substituting $z = \lambda/c$ and $\mathrm{d}\lambda/\mathrm{d}z = c$ leads to

$$= \int_0^1 \frac{e^{-zc}(zc)^x}{x!c} \frac{(z)^{\alpha-1}(1-z)^{\beta-1}}{B(\alpha,\beta)} \frac{1}{c}\mathrm{dz}$$
$$= \frac{c^x\Gamma(\alpha+\beta)}{x!\Gamma(\alpha)\Gamma(\beta)} \int_0^1 e^{-zc}(z)^x(z)^{\alpha-1}(1-z)^{\beta-1}\mathrm{dz}.$$

${}_1F_1$ is the confluent hypergeometric function of first order (Definition A.6). With this, we get

$$f_{\mathrm{PB}}(x|\alpha,\beta,0,c) = \frac{c^x\Gamma(\alpha+\beta)\Gamma(\alpha+x)}{\Gamma(x+1)\Gamma(\alpha)\Gamma(\alpha+\beta+x)}\ {}_1F_1(\alpha+x;\alpha+\beta+x;-c) \quad \text{for } x \in \mathbb{N}_0. \tag{3.1}$$

A random variable $X \sim \mathrm{PB}(\alpha,\beta,0,c)$ has expectation and variance

$$\mathrm{E}_{\mathrm{PB}(\alpha,\beta,0,c)}[X] = \mathrm{E}_{\mathrm{Beta}(\alpha,\beta,0,c)}[\lambda] = \frac{\alpha c}{\alpha+\beta} \qquad \text{and}$$

$$\mathrm{Var}_{\mathrm{PB}(\alpha,\beta,0,c)}[X] = \mathrm{E}_{\mathrm{Beta}(\alpha,\beta,0,c)}[\lambda] + \mathrm{Var}_{\mathrm{Beta}(\alpha,\beta,0,c)}[\lambda] = \frac{\alpha c}{\alpha+\beta} + \frac{\alpha\beta c^2}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

Another compound Poisson distribution that will show up in this thesis is the Poisson-inverse Gaussian (PIG) distribution. In contrast to the PB distribution the PIG is more popular.

Example 3.5 (Poisson-Inverse Gaussian (PIG) Distribution) *The PMF of the (compound) Poisson-inverse Gaussian distribution is a Poisson distribution whose intensity parameter itself follows an inverse Gaussian distribution (Definition A.2). The PMF of the PIG distribution can be derived and is given in (Holla, 1967) by*

$$f_{PIG}(x|\mu,\lambda) = \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!} f_{\mathrm{IG}}(\lambda|\mu,\lambda)\mathrm{d}\lambda$$
$$= \left(\frac{2\lambda}{\pi}\right)^{\frac{1}{2}} \frac{1}{x!} e^{\frac{\lambda}{\mu}} \left[\frac{\lambda}{2\left(1+\frac{\lambda}{2\mu^2}\right)}\right]^{\frac{1}{2}\left(x-\frac{1}{2}\right)} K_{x-\frac{1}{2}}\left(\sqrt{2\lambda\left(1+\frac{\lambda}{2\mu^2}\right)}\right).$$

Its probability generating function (PGF) is given by

$$G_{PIG}(z|\mu,\lambda) = \exp\left(-\frac{\mu}{\lambda}\left[[1+2\beta(1-z)]^{\frac{1}{2}}-1\right]\right).$$

A random variable $X \sim \mathrm{PIG}(\mu,\lambda)$ has expectation and variance

$$\mathrm{E}_{\mathrm{PIG}(\mu,\lambda)}[X] = \mu \qquad \text{and} \qquad \mathrm{Var}_{\mathrm{PIG}(\mu,\lambda)}[X] = \mu + \frac{\mu^3}{\lambda}.$$

Note, that sometimes the PIG distribution is parameterized via (μ,σ) where $\sigma = \frac{\mu}{\lambda}$ is the dispersion parameter (see Rigby et al., 2019).

Example 3.6 (Poisson-shifted Gamma (PsG) Distribution) *The PMF of the (compound) Poisson distribution where the intensity parameter itself follows a shifted gamma distribution, i. e. a gamma distribution with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$ that is shifted by some value $\lambda > 0$ is given by*

$$f_{\mathrm{PsG}}(x|\alpha, \beta, \lambda) = \sum_{i=0}^{x} \frac{\Gamma(\alpha+i)\beta^i \lambda^{x-i} e^{-\lambda}}{\Gamma(\alpha) i! (1+\frac{1}{\beta})^{\alpha+i} (x-i)!}$$

A random variable $X \sim \mathrm{PsG}(\alpha, \beta, \lambda)$ has expectation and variance

$$\mathrm{E}_{\mathrm{PsG}}[X] = \lambda + \frac{\alpha}{\beta} \qquad \textit{and} \qquad \mathrm{Var}_{\mathrm{PsG}}[X] = \lambda + \frac{\alpha(1+\beta)}{\beta^2}.$$

This distribution is also called Delaporte distribution (see Definition A.10) with parameters μ, σ and ν, where $\mu = \lambda + \frac{\alpha}{\beta}$, $\sigma = \frac{1}{\alpha}$ and $\nu = \frac{\lambda}{\lambda + \frac{\alpha}{\beta}}$.

3.1.3 Convolution of Distributions

In case only the sum of several latent observations can be measured and needs to be modeled, the resulting distribution must be determined. In other words, we need the probability distribution of $Y = X_1 + \cdots + X_n$ which is the convolution of the individual probability distributions of the latent observations $X_1, \ldots, X_n$.

Definition 3.4 (Convolution of Distributions) *Let Y be the sum of n independent random variables $X_1, \ldots, X_n$ following distributions $\mathcal{D}_1, \ldots \mathcal{D}_n$ with PDFs or PMFs $f_1, \ldots, f_n$ and parameters $\theta_1, \ldots, \theta_n$. Then in case the $\mathcal{D}_i$ are continuous distributions, the PDF of Y is given by*

$$f_Y(y|\boldsymbol{\theta}) = \int_0^y \int_0^{y-x_1} \cdots \int_0^{y-\sum_{j=1}^{n-2} x_j} f_1(x_1|\theta_1) f_2(y-x_1|\theta_2) \cdots f_n\left(y - \sum_{j=1}^{n-1} x_j \middle| \theta_n\right) dx_{n-1} \cdots dx_2 dx_1.$$

In case the $\mathcal{D}_j$ are discrete distributions, the PMF of Y is given by

$$f_Y(y|\boldsymbol{\theta}) = \sum_{x_1=0}^{y} \sum_{x_2=0}^{y-x_1} \cdots \sum_{x_{n-1}=0}^{y-\sum_{j=1}^{n-2} x_j} f_1(x_1|\theta_1) f_2(y-x_1|\theta_2) \cdots f_n\left(y - \sum_{j=1}^{n-1} x_j \middle| \theta_n\right).$$

In the case of $Y = X_1 + X_2$ this simplifies in the continuous case to

$$f_Y(y|\boldsymbol{\theta}) = \int_0^y f_1(x_1|\theta_1) f_2(y-x_1|\theta_2) dx_1, \tag{3.2}$$

and in the discrete case to

$$f_Y(y|\boldsymbol{\theta}) = \sum_{x_1=0}^{y} f_1(x_1|\theta_1) f_2(y-x_1|\theta_2). \tag{3.3}$$

Deriving the resulting probability distribution for Y is not straightforward and often it is impossible to calculate it explicitly. Therefore, approximation methods must often be used. However, there are special cases where the resulting distribution can be calculated, since it is again a known parametric distribution. Next, we will give some examples of these cases and present the convolutions used in this thesis. In general it is easier to calculate the convolution if the f_j come from the same distribution family and differ only in their parameters θ_i. We start with distributions that are infinitely divisible, i. e. a random variable Y following a probability distribution $\mathcal{D}$ that can be divided for each positive integer n in n i.i.d. random variables with $Y = X_1 + \cdots + X_n$ (see Sato, 1999). Clearly the normal distribution is such an infinitely divisible distribution. We start with the convolution of normally distributed random variables, where the resulting convolution is itself a normal distribution. This is true for all possible parameters of the original normal distributions.

Example 3.7 (Convolution of $\mathrm{Normal}(\mu_j, \sigma_j^2)$ Distributions) *The normal distribution is one of the rare distributions where the random variables do not have to follow the same normal distribution i. e. the $\theta_j = (\mu_j, \sigma_j^2)$ may differ, so that their sum still follows a normal distribution. In detail, if the independent $X_j \sim \mathrm{Normal}(\mu_j, \sigma_j^2)$ then $Y = X_1 + \cdots + X_n$ follows a $\mathrm{Normal}(\mu_1 + \cdots + \mu_n, \sigma_1^2 + \cdots + \sigma_n^2)$ distribution.*

Next, we look at random variables that need to share some common parameter in order to maintain the parametric distribution for their convolution.

Example 3.8 (Convolution of $\mathrm{NB}(r_j, p)$ Distributions) *The convolution of n NB distributions that share the same p parameter is again a NB distribution. In detail, if the independent $X_j \sim \mathrm{NB}(r_j, p)$ (see Definition A.8) then $Y = X_1 + \cdots + X_n$ follows a $NB(r_1 + \cdots + r_n, p)$ distribution (see e.g. Furman, 2007).*

Some distributions require that all X_j come from exactly the same distribution, i. e. $\theta_i = \theta_j$ to maintain the parametric distribution for the convolution. Trivially, this is the case for both normal and NB distributions. Here we will present the convolution of exponentially distributed random variables, since we will need this in Chapter 5.

Example 3.9 (Convolution of $\mathrm{EXP}(\lambda)$ Distributions) *The sum Y of independent exponentially distributed random variables (Definition A.4) with the same intensity parameter λ follows an Erlang distribution (see Feldman and Valdez-Flores, 2010). An Erlang distribution is a gamma distribution (Definition A.3) with integer shape parameter $\alpha = n$, which represents the number of exponentially distributed summands, i. e. $\mathrm{Gamma}(n, \lambda)$. Note that the exponential distribution itself is a special case of the gamma distribution, with shape parameter $\alpha = 1$. Taken together, this means if $X_j \sim \mathrm{EXP}(\lambda) = \mathrm{Gamma}(1, \lambda)$, then $Y = X_1 + \cdots + X_n \sim \mathrm{Gamma}(n, \lambda)$.*

As mentioned before, not all distributions maintain their distribution after convolution. However, for the convolution of the NB distribution with different parameters r_i and p_i the resulting distribution can still be described. We use this result in Chapter 5.

Example 3.10 (Convolution of $\mathrm{NB}(r_j, p_j)$ Distributions) *Furman (2007) shows that f_Y describing the PMF of the convolution $Y = X_1 + \ldots + X_n$ of n independent random variables $X_j \sim \mathrm{NB}(r_j, p_j)$ (see Definition A.8), is given by*

$$f_Y(y|\boldsymbol{r}, \boldsymbol{p}) = f_{n\text{-}NB}(y|\boldsymbol{r}, \boldsymbol{p}) = R \sum_{k=0}^{\infty} \delta_k f_{NB}(y|r + k, p_{max}) \quad \text{for } y \in \mathbb{N}_0, \tag{3.4}$$

where $r = r_1 + \cdots + r_n$ and $p_{max} = \max_j\{p_j\}$.

$$R = \prod_{j=1}^{n} \left(\frac{(1 - p_j)p_{max}}{(1 - p_{max})p_j} \right)^{r_j} \qquad \text{and}$$

$$\delta_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} \sum_{j=1}^{n} r_j \left(1 - \frac{(1 - p_{max})p_j}{(1 - p_j)p_{max}} \right)^{i} \delta_{k+1-i},$$

for $k = 0, 1, \ldots$ with $\delta_0 = 1$.
This distribution can also be described as Y following a compound NB distribution whose size parameter itself follows some discrete distribution, i. e.
$Y \sim \mathrm{NB}(r + K, p_{max})$ with $\mathrm{P}(K = k) = R\delta_k$, for $k = 0, 1, \ldots,$ where R and δ_k are given above.

In Chapter 5 we use the convolution of lognormally distributed random variables. Even if all parameters are equal, the convolution of lognormals has no analytic form. In contrast to the convolution of NB distributions, as far as we know, the resulting distribution of the convolution of lognormals can not be written analytically down. However, it is possible to apply the method by Fenton (1960) to approximate the resulting distribution.

Example 3.11 (Convolution of $\mathrm{LN}(\mu_j, \sigma_j^2)$ Distributions) *The PDF f_Y of $Y = X_1 + \ldots + X_n$ with independent $X_j \sim \mathrm{LN}(\mu_j, \sigma_j^2)$ is approximated by using the method by Fenton (1960), i. e. the resulting convolution is approximated by the distribution of a random variable $B \sim \mathrm{LN}(\mu_B, \sigma_B^2)$ such that*

$$\mathrm{E}(B) = \mathrm{E}(X_1 + \ldots + X_n) \quad \text{and} \quad Var(B) = Var(X_1 + \ldots + X_n).$$

According to the expectation and variance of the lognormal distribution (see Definition A.1) it follows that μ_B and σ_B are chosen such that the following equations are fulfilled:

$$\exp\left(\mu_B + \frac{\sigma_B^2}{2}\right) = \exp\left(\mu_1 + \frac{\sigma_1^2}{2}\right) + \ldots + \exp\left(\mu_n + \frac{\sigma_n^2}{2}\right) =: \Gamma$$

and

$$\begin{aligned}&\exp\bigl(2\mu_B + \sigma_B^2\bigr)\bigl(\exp\left(\sigma_B^2\right) - 1\bigr) = \\ &\exp\left(2\mu_1 + \sigma_1^2\right)\bigl(\exp\left(\sigma_1^2\right) - 1\bigr) + \ldots + \exp\left(2\mu_n + \sigma_n^2\right)\bigl(\exp\left(\sigma_n^2\right) - 1\bigr) =: \Delta.\end{aligned}$$

That is achieved by choosing

$$\mu_B = \log(\Gamma) - \frac{1}{2}\sigma_B^2 \quad \textit{and} \quad \sigma_B^2 = \log\left(\frac{\Delta}{\Gamma^2} + 1\right).$$

In these examples we have always convolved distributions of the same distribution family. If we want to determine the convolution of different distributions we have to calculate the integral or the sum given in Definition 3.4. In particular, if we are convolving only two distributions, the integral (3.2) or the sum 3.3 should not be too difficult to calculate using mathematical software such as R or MATLAB.
An exception from this necessity could occur if different compound Poisson distributions (Definition 3.3) are to be convoluted. Karlis and Xekalaki (2005) show that a convolution of a compound Poisson distribution with another compound Poisson distribution leads again to a compound Poisson distribution. More precisely, the new intensity density equals the convolution of the original intensity densities.

Definition 3.5 (Convolution of Compound Poisson Distributions) *The sum $Y = X_1 + X_2$ of two compound Poisson random variables, where $X_1 \sim \text{Comp-Pois}_{f_1}(\theta_1)$ and $X_2 \sim \text{Comp-Pois}_{f_2}(\theta_2)$, itself follows a compound Poisson distribution, where the intensity density is the convolution of the two original intensity densities f_1 and f_2.*

Therefore, a convolution of a Poisson distribution with some compound Poisson distribution leads in turn to a compound Poisson distribution in which the intensity density is the intensity density of the original compound distribution but is shifted by the parameter of the Poisson distribution. This is used in the following example, where a NB distribution and a Poisson distribution are convoluted.

Example 3.12 (Convolution of $\text{NB}(r, p)$ and $\text{Pois}(\lambda)$ Distribution) *The convolution of a Poisson distribution and a NB distribution results in a PsG distribution, i. e. a compound Poisson distribution with a shifted gamma intensity density, see Example 3.6. This resulting distribution is also called Delaporte distribution, see Definition A.10. In detail, if $X_1 \sim \text{NB}(r, p)$ and $X_2 \sim \text{Pois}(\lambda)$, then $Y = X_1 + X_2 \sim \text{PsG}\left(r, \frac{p}{1-p}, \lambda\right) = \text{DEL}\left(\lambda + \frac{r(1-p)}{p}, \frac{1}{r}, \frac{\lambda}{\lambda + \frac{r(1-p)}{p}}\right)$.*

Note, that in Chapter 4 we need distributions that are not only infinitely divisible, but self-decomposable.

Definition 3.6 (Self-decomposable Distribution) *Let $\hat{\mu}$ be the characteristic function of a random variable X following the one-dimensional law $\mathcal{D}$. $\mathcal{D}$ is self-decomposable iff*

$$\hat{\mu}(z) = \hat{\mu}(cz)\hat{\mu}_c(z)$$

for all $z \in \mathbb{R}$ and all $c \in (0, 1)$ and some family of characteristic functions $\{\hat{\mu}_c : c \in (0, 1)\}$.

Hence, self-decomposable distributions are a subclass of infinitely divisible distributions.

3.2 Stochastic Processes

In Chapter 4 we will show how to connect steady state distribution with stochastic processes of gene transcription. The approach that we will present uses a special form of stochastic differential equations (SDEs) involving a stochastic process $(X_t)_{t \in T}$, called Ornstein-Uhlenbeck (OU) process. Here we will give some background information that is needed to follow that chapter. Although there exist many stochastic processes, we will only focus on Markov, Lévy and OU processes. References can be found among others in Kijima (1997),Barndorff-Nielsen and Shephard (2001b), Barndorff-Nielsen and Shephard (2001a), Sato (1999) and Rogers and Williams (2000). A Markov process is a stochastic process that satisfies the Markov property.

Definition 3.7 (Markov Process) *A process $(X_t)_{t \geq 0}$ is called a Markov process if, for each n and every $i_0, \ldots, i_n$ and $j \in \mathbb{N}$ the Markov property holds, i. e.*

$$\mathcal{P}(X_{n+1} = j | X_0 = j_0, \ldots, X_n = j_n) = \mathcal{P}(X_{n+1} = j | X_n = j_n),$$

with $P(X = x) > 0$.

Markov processes can be subdivided in different types, depending on a discrete or continuous state space and on the time parameter being discrete or continuous. Often discrete-time Markov processes are called Markov chains and continuous-time Markov processes with discrete state space are called Markov jump processes. However, in this thesis we only refer to a Markov process that runs on discrete state spaces. All birth-death processes and queuing systems in Chapter 4 and Appendix E are such Markov jump processes. Lévy processes fulfill the Markov property and are therefore a subset of Markov processes.

Definition 3.8 (Lévy Process) *A process $(X_t)_{t \geq 0}$ with values in $\mathbb{R}^d$ is called a Lévy process (or process with stationary independent increments) if it has the following properties:*

- *For almost all ω in the considered probability space, the mapping $t \mapsto X_t(\omega)$ is right-continuous on $[0, \infty]$,*
- *for $0 \leq t_0 < t_1 < \cdots < t_n$, the random variables $Y_j := X_{t_j} - X_{t_{j-1}}$ $(j = 1, \ldots, n)$ are independent,*
- *the law of $X_{t+h} - X_t$ depends on $h > 0$, but not on t.*

One famous example for a Lévy process is the Brownian motion. However, we will only use non-Gaussian increasing Lévy processes. Additionally we are only looking at Lévy processes with positive increments.

Definition 3.9 (Subordinator) *An increasing Lévy process is called a subordinator.*

In the following we will introduce the non-Gaussian subordinators that we will use in this thesis.

Definition 3.10 (Poisson Process) *A Poisson process X_t with intensity parameter λ starts almost surely in zero and for all $0 \leq s < t$ one has independent $X_t - X_s \sim Pois((t-s)\lambda)$.*

The just defined Poisson process is also called Poisson point process, as it just marks time points when events happen. This is used to construct the following compound Poisson process.

Definition 3.11 (Compound Poisson Process, CPP) *A compound Poisson process (CPP) Z_t with intensity parameter λ is defined as*

$$Z_t = \sum_{i=1}^{N_t} Y_i,$$

where N_t is a Poisson process with parameter λ, and the jumps Y_i are independent and identically distributed random variables. The characteristic function of a CPP depends on the distribution of the Y_i and is given by

$$\hat{\mu}_{Z_t}(z) = \exp(t\,\lambda(\hat{\mu}_Y(z) - 1)),$$

where $\hat{\mu}_Y$ is the characteristic function of the Y_i.

Another type of increasing Lévy process is the inverse Gaussian (IG) process. An (IG) process X_t can be defined in several ways. Originally it was defined through the first-passage time of a Gaussian process (Applebaum, 2004). (Ye and Chen, 2014) showed how to define the inverse Gaussian process analogously to the well known gamma process by its increments.

Definition 3.12 (Inverse Gaussian (IG) Process) *The IG process $X_t, t \geq 0$ is a stochastic process characterized by the following properties:*

- *X_t has independent increments, i. e. $X_{t_2} - X_{t_1}$ and $X_{t_4} - X_{t_3}$ are independent for all $t_4 > t_3 \geq t_2 > t_1$.*
- *$X_t - X_s$ follows an IG distribution (see Definition A.2) $\mathrm{IG}(M(t) - M(s), \eta[M(t) - M(s)]^2)$, for all $t > s \geq 0$,*

where $\eta > 0$ and $M(t)$ is a monotone increasing function of the process. In detail $M(t)$ is the mean function of the process. The variance function is given by $\eta M(t)$ and therefore is also a monotone increasing function.

In case $M(t)$ is a linear function, the distribution of increments is only dependent on the time step $t - s$ and hence Definition 3.8 directly implies that the inverse Gaussian process is a Lévy process. Since it is increasing it is a subordinator (Definition 3.9). In fact the IG process is a pure jump Lévy process and can be seen as a limiting CPP, where the jump arrival rate goes to infinity and the jump sizes, that follow a specific IG distribution go to zero.

The main part of theory that we will use in Chapter 4 centers around Ornstein-Uhlenbeck (OU) processes. Their construction using a Lévy process will be defined next.

Definition 3.13 (Ornstein-Uhlenbeck (OU) Process) *Following Barndorff-Nielsen and Shephard (2001b), an Ornstein-Uhlenbeck (OU) process y_t is the solution of a stochastic differential equation (SDE) of the form*

$$\mathrm{d}y_t = -\lambda y_t \,\mathrm{d}t + \mathrm{d}z_t, \tag{3.5}$$

where z_t, with $z_0 = 0$ almost surely, is a Lévy process (see Definition 3.8). If the Lévy process has no Gaussian components, the process z_t is called a non-Gaussian OU process or also a process of OU-type. Often, this is also shortened to OU process. Barndorff-Nielsen and Shephard (2001a) also call z_t a background-driving Lévy process as it drives the OU process.

In Chapter 4, we will only use OU processes that are driven by non-Gaussian subordinators z_t, that were introduced above.

3.3 Parameter Inference

Whenever a parametric model needs to be fitted to given data, parameter inference is needed. As mentioned before this is a huge field and a lot of literature and methods exist. Here, we will only list the methods we need in this thesis. In general there exist two different ways to perform parameter inference: the frequentist and the Bayesian approach. Both need the likelihood of the model.

Given some data $\boldsymbol{x}$ of length k and some model f with parameter vector $\boldsymbol{\theta}$ the likelihood function L is given by

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{i=1}^{k} f\left(x_i|\boldsymbol{\theta}\right). \tag{3.6}$$

3.3.1 Maximum Likelihood Estimation

The traditional frequentist approach uses maximum likelihood (ML) inference, meaning to find an estimate of the parameter $\boldsymbol{\theta}$ for which the likelihood function is maximized. This is the same as maximizing the log-likelihood function ℓ of the model parameters, given by

$$\ell(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{k} \log\left[f\left(x_i|\boldsymbol{\theta}\right)\right]. \tag{3.7}$$

Since often it is easier to find a minimum value than a maximum value, often the negative log-likelihood function is minimized. The resulting $\hat{\theta}$ is called maximum likelihood estimator (MLE). There exist many different algorithms to perform ML estimation. In this thesis, we use the Nelder-Mead (Nelder and Mead, 1965) algorithm – downhill simplex method – or the BFGS (Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970) algorithm – a quasi Newton method. These are often used standard algorithms. For more information we refer among many others to Rossi (2018).

3.3.2 Bayesian Parameter Inference

Bayesian methods in general consider the model parameter not as a fixed value that needs to be determined but as a random variable that follows a distribution. Therefore Bayesian parameter inference determines the distribution of the parameter. Bayesian inference is computationally more expensive. With increasing computational power its usage has become increasingly popular over the last 30 years. The core of Bayesian inference is the computation of the posterior probability of the model parameter $\boldsymbol{\theta}$ given the data via Bayes' theorem:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}, \tag{3.8}$$

where for independent x_i, $p(\boldsymbol{x}|\boldsymbol{\theta})$ is given by the likelihood function of the parametric model, see Equation (3.6). Markov chain Monte Carlo (MCMC) methods use Equation (3.8) to accept or decline proposed parameter values and move around in the parameter space to generate a sample from the posterior distribution. Using MCMC chains, Bayesian credibility intervals and other properties such as mean and quantiles of the sample can be calculated. For more information see Lee (2012) and Gelman et al. (2013).

However, these sampling schemes can be computationally more demanding than optimization which is why we use in Section 5.5 an optimized sampling scheme. This Hamiltonian Monte Carlo (HMC)-based algorithm called No-U-Turn sampler (NUTS, Hoffman and Gelman, 2014) is implemented in the programming language Stan through its interface **RStan** (Stan Development Team, 2019). HMC (also known as hybrid Monte Carlo) is an MCMC method that works similar to the Metropolis-Hastings algorithm. The main difference is the proposal and acceptance of new parameters. Using the evolution of Hamiltonian dynamics, new propositions are generated via the so-called leapfrog integrator, i. e. a new parameter set is proposed after L intermediate step updates of the old parameter set using so-called momentum variables, that are also updated in each intermediate step. The step size of these intermediate steps is given by the step size parameter ϵ. This procedure leads to new parameter proposals that are more likely to be accepted than standard Metropolis propositions generated by Gaussian random walks. Since HMC proposals move more in space, HMC chains propose less correlated parameter sets. Therefore fewer samples (which results in shorter chains), are sufficient to approximate the posterior distributions.

In contrast to the original HMC, NUTS does not require the specification of the number of leapfrog steps L. In addition, the Stan (and thus **Rstan**) implementation tunes the step size ϵ in an automated manner.

3.4 Model Selection and Goodness of Fit

If several models are available for data fitting, an objective criterion is needed to select the model that fits the data best. To avoid over-fitting, it is advisable to use a criterion that penalizes more complicated models. In this thesis we use the Bayesian information criterion (BIC, Schwarz, 1978) which is given by

$$\mathrm{BIC}(\hat{\boldsymbol{\theta}}) = -2\ell(\hat{\boldsymbol{\theta}}) + \log(k)\dim(\hat{\boldsymbol{\theta}}), \tag{3.9}$$

where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood parameter estimate and $\ell()$ the log-likelihood function, see (3.7), of the respective model. $\dim(\hat{\boldsymbol{\theta}})$ is the number of model parameters and k the size of the dataset. Taken together, the model with the smallest BIC is considered most appropriate among all models considered.

In practice, only the best available model is selected during model selection. This does not necessarily mean that the model really matches the data. This goodness of fit can be assessed via statistical hypothesis testing. In the case of fitting parametric distributions to datsets, we therefore want to check whether the dataset X really follows this specific distribution, i. e. whether it has a probability distribution F_0. This leads to the null hypothesis:

$$H_0 : X \sim F_0.$$

The χ^2 test from Pearson (1900), which we will use in this thesis, bins the observations and compares their numbers per bin with the expected frequency of each bin given by the probability distribution. The test statistic that described the deviation of those two frequencies then follows a χ^2 distribution.

Before applying this test, the observations and the probability distribution to test must be prepared. This means that the data and the probability distribution must be binned in the same way. When this test is used for discrete distributions, the discrete counts of the distribution are a natural choice of bins. Since a finite number of bins is needed, we choose a threshold above which all observations are combined to one final bin. In Chapter 4 we use our own implementation of this goodness-of-fit test for the proposed distributions, since they are not all available by default in already existing packages of the χ^2 test.

4 Modeling Single-Cell mRNA Counts

In this chapter, we describe how to link mechanistic processes which describe possible models of mRNA transcription to steady state distributions that are used to model single-cell mRNA counts. The chapter is based on and is partly identical to the following preprint:

> **Amrhein, L.**, Harsha, K., and Fuchs, C. (2019). A mechanistic model for the negative binomial distribution of single-cell mRNA counts. *bioRxiv 657619.*

Picking a distribution is crucial for data modeling and analysis. There are two ways to approach this decision. Traditionally, an underlying mechanistic model of the mRNA transcription process is selected and steady state distributions of the molecules involved are calculated. In the first part of this chapter we will show how this is done for the most common models in literature. Afterwards we will focus on the second approach: A distribution is selected based on its data fit. We will show how a possible underlying mRNA transcription model can be derived from this.

4.1 Prior Work

When analyzing the outcomes of single-cell RNA sequencing (scRNA-seq) experiments, it is essential to adequately consider the properties of the resulting data. Many methods assume a parametric distribution for the sequencing counts due to its larger power compared to non-parametric approaches. To that end, a family of parametric distributions needs to be chosen which adequately models the data. This is particularly important for users since the distribution of a selected tool directly impacts preprocessing.

Count data is most accurately described by discrete distributions (see Appendix A.2) unless count numbers are without exception very high in which case continuous distributions (see Appendix A.1) might also be suitable.

A commonly chosen count distribution is the Poisson distribution (Definition A.7), which can be derived from a simple birth-death model of mRNA transcription and degradation. However, due to widespread overdispersed data, which means that the observed variance is higher than explained by the model, a Poisson distribution is seldom suitable. Another typical choice is a three-parameter PB distribution (see Example 3.4, Delmans and Hemberg, 2016, Vu et al., 2016) which can be derived from a DNA switching model (also called *random telegraph model*, see Dattani and Barahona, 2017, or *basic model of gene activation and inactivation*, see Raj et al., 2006). This mechanistic description is most often used when transcription processes are discussed (e. g. Buettner et al., 2015, Jansen and Pfaffelhuber, 2015). Parameters of the PB distribution can be estimated from scRNA-seq data (Kim and Marioni, 2013), as well as experimentally measured and inferred (Suter et al., 2011). This distribution provides good estimates of scRNA-seq data; however, it entails the estimation of three parameters which introduces a high computational cost (Kim and Marioni, 2013). A frequent third choice is the NB distribution, used by several tools that analyze single-cell gene expression measurements such as SCDE (Kharchenko et al., 2014), Monocle 2 (Qiu et al., 2017) and many more (see Table 4.1). This distribution is chosen due to computational convenience and good empirical fits. Some papers already derived the NB distribution as steady state distribution for either mRNA or protein numbers. For mRNA distributions, the NB was only derived by considering it as asymptotic steady state distribution of the switching model (see Raj et al., 2006). However, our discussion shows that this will entail biologically unrealistic assumptions. However, our discussion shows that this will entail biologically unrealistic assumptions. Others like Shahrezaei and Swain (2008) extend the basic model by adding the step of protein translation and inferred the distribution of those proteins. When mRNA degrades much faster than proteins, this distribution converges in steady state to a NB distribution. In this case protein translation can be described as instantaneous bursts that follow a geometric distribution. The derivation of geometric bursts of proteins and a NB distribution as their steady state distribution was already described by Berg (1978) and Paulsson et al. (2000). In theory, this mechanism can be transferred to mRNA transcription, but as far as we know, it has not yet been explicitly formulated.

Table 4.1 provides an overview of computational tools to analyze scRNA-seq data and their distributional assumptions.

Tool	**Category**	**Distribution Model**					**Notes**
		NB	PB	Other	ZI	Hurdle	
BASiCS	• Normalization • Differential Expression • Variable Genes • Simulation	⊗	○	○	○	○	Poisson-gamma, Bayesian hierarchical models, Vallejos et al. (2015)
bayNorm	• Normalization • Imputation • Simulation	⊗	○	○	○	○	Binomial sampling with NB priors, Tang et al. (2020)

BEAM	• *Ordering* • *Expression Patterns* • *Differential Expression*	✗	○	○	○	○	Branch-dependent gene expression as a contrast between two NB GLMs, Qiu et al. (2017)
BPSC	• Differential Expression	○	✗	○	✗	○	BP3 is PB; BP4 adds fractions scaling parameter; BP5 adds ZI; Vu et al. (2016)
ComBat	• *Batch Correction*	○	○	✗	○	○	Uses normal distribution on normalized data, Stein et al. (2015)
DCA	• Imputation	✗	○	○	✗	○	Eraslan et al. (2019)
DPT	• Ordering • Expression Patterns • Visualization	○	○	✗	○	✗	Normal distribution, Haghverdi et al. (2016)
D3E	• Differential Expression	○	✗	○	○	○	Delmans and Hemberg (2016)
diffxPy	• *Differential Expression*	✗	○	○	✗	○	`https://github.com/theislab/diffxpy`
limma	• *Normalization* • *Differential Expression* • *Gene Sets* • *Batch Correction*	○	○	✗	○	○	Linear model using normal distributions, Ritchie et al. (2015)
lineagePulse	• Differential Expression • Expression Patterns • Visualization • Simulation	✗	○	○	✗	○	`https://github.com/YosefLab/LineagePulse`
MAST	• Quality Control • Normalization • Differential Expression • Gene Sets • Gene Networks	○	○	○	○	✗	Logistic regression & Gaussian linear model for expressed genes, Finak et al. (2015)
M3Drop	• Differential Expression • Marker Genes • Visualization • Simulation	✗	○	○	○	○	Depth-adjusted NB, Andrews and Hemberg (2018)
powSimR	• Visualization • Simulation	✗	○	○	✗	○	The user has the option to include zero-inflation (default is not to use it), Vieth et al. (2017)
SAVER	• Imputation	✗	○	○	○	○	Huang et al. (2018)

SCDE	• Differential Expression • Gene Sets • Visualization	○	○	⊗	⊘	○	Poisson-NB mixture: Poisson for dropout, NB for amplified expression, Kharchenko et al. (2014)
SCHiRM	• Normalization • Gene Networks • Visualization • Simulation	○	○	⊗	○	○	Poisson-lognormal, Intosalmi et al. (2018)
scImpute	• Imputation	○	○	⊗	⊘	○	Gamma-normal mixture on log-transformed expression: dropouts modeled via normal distribution, Li and Li (2018)
sctransform	• Normalization • Integration • Differential Expression • Transformation • Visualization	⊗	○	○	○	○	Regularized NB regression, Hafemeister and Satija (2019)
scVI	• Dimensionality Reduction	⊗	○	○	⊗	○	ZINB-like generative model, Lopez et al. (2018)
Splatter	• Visualization • Simulation	⊗	○	○	⊗	○	Some intermediate steps; gene- and cell-wise mean are modeled with gamma distribution, Zappia et al. (2017)
ZIFA	• Dimensionality Reduction	○	○	⊗	⊗	○	Zero-inflated Gaussian (Bernoulli-normal mixture), Pierson and Yau (2015)
ZINB-WaVE	• Normalization • Dimensionality Reduction • Simulation	⊗	○	○	⊗	○	Risso et al. (2018)

Table 4.1: Overview of single-cell analysis tools with underlying distributional assumptions (black ticks, gray ticks for available alternative assumption). Categories of the tools are taken from `www.scrna-tools.org`. Tools that were not listed are categorized by us. Those categories are written in italics. Details on the categories are listed in Appendix C.

Among the 23 listed tools, 13 use a negative binomial (NB) distribution and two a Poisson-beta (PB) distribution. In eleven tools a zero-inflated distribution is implemented to take into account the more than expected zeros. This can overlap with the cases before as these models extend a parametric distribution by adding a dropout part. We will discuss zero-inflation and its integration into models later in Section 4.5.1 when we will discuss real-world data and suitable models.

4.2 Inferring Distributions from Gene Expression Mechanisms

The aim of this section is to show how to infer a mechanistic transcription process after selecting a probability distribution for mRNA counts. We will get there by reversing the traditional way. Therefore we will sketch first how steady state distributions can be inferred from the most common mechanistic processes and how this can be generalized. These models describe the number of mRNA molecules in a cell for either *one* gene or for a group of genes for which we can assume identical kinetic parameters. We start with the simple birth-death model, that is later generalized so that we obtain a general approach how to infer steady state distributions for this type of models. Alterations in the transcription and degradation model lead to alterations in the resulting mRNA count distribution. The well known switching model fits into this generalized form, and its resulting steady state distribution will be put into context.

4.2.1 Basic Model: Constant Transcription and Degradation

The simple birth-death model – we will call it basic model – for mRNA transcription and degradation (Dattani and Barahona, 2017, Peccoud and Ycart, 1995) in which transcription and degradation occur at constant rates r_{tran} and r_{deg} is shown in Figure 4.1.

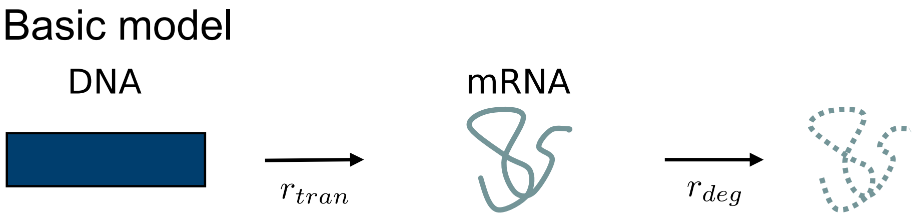


Figure 4.1: Basic model of gene expression consists of constant transcription and constant degradation of mRNA.

Although the steady state distribution of this simple model can be easily inferred, we included the complete derivation in Appendix D.1. This is particularly important not only because it forms the basis for all subsequent sections, but also to introduce consistent notation. In short, the master equation

$$\frac{d\mathcal{P}(n,t)}{dt} = r_{tran}\mathcal{P}(n-1,t) + r_{deg}(n+1)\mathcal{P}(n+1,t) - (r_{tran} + r_{deg}\, n)\mathcal{P}(n,t) \quad (4.1)$$

leads to the probability generating function of the number of molecules at time point t, given by

$$G(z,t|n_0) = \left[(z-1)e^{-r_{deg}t} + 1\right]^{n_0} e^{I(t)(z-1)}, \text{ with } I(t) = \frac{r_{tran}}{r_{deg}}\left(1 - e^{-r_{deg}t}\right).$$

This is the probability generating function of the number of mRNA molecules at time t with initial number of mRNA molecules n_0. In steady state, i. e. $t \to \infty$ the first part disappears and thus the steady state distribution is independent of the starting material of molecules. The second part, which corresponds to the probability generating function of a compound Poisson distribution 3.3 with intensity function $I(t)$, is simplified to a Poisson distribution with time-independent, constant intensity parameter $I = \frac{r_{tran}}{r_{deg}}$. In summary, the steady state distribution of mRNA molecules is described in the basic model by a Poisson distribution with intensity parameter $\frac{r_{tran}}{r_{deg}}$. Note that the basic model can be understood as a queuing system (Adan and Resing, 2015) where costumers ($\widehat{=}$ mRNA molecules) arrive, wait until they are called and then served ($\widehat{=}$ mRNA molecule is present in the cell) until they finally leave the system ($\widehat{=}$ mRNA molecule degrades). Further calculations (see Appendix E.1) show that this results in the same Poisson distribution.

4.2.2 Generalization: Time-Varying Transcription and Constant Degradation

The basic model can be generalized, so that in the general context, we consider a transcription-degradation model with stochastic time-varying transcription rate R_t and deterministic constant degradation rate r_{deg} as shown in Figure 4.2.

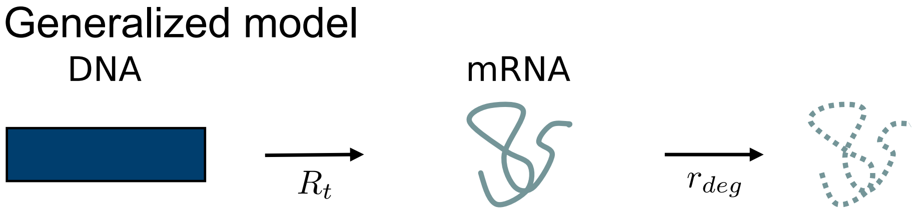


Figure 4.2: Generalized model of gene expression consists of transcription governed by some time-varying transcription rate R_t and constant degradation of mRNA.

The derivation is completely identical to the one for the basic model, only the transcription rate is now described by the stochastic time-varying rate R_t. Therefore the master equation reads

$$\frac{d\mathcal{P}(n,t)}{dt} = R_t\mathcal{P}(n-1,t) + r_{deg}(n+1)\mathcal{P}(n+1,t) - (R_t + r_{deg}\, n)\mathcal{P}(n,t), \qquad (4.2)$$

and the moment generating function of mRNA counts is analogously given by

$$G(z,t|n_0) = \left[(z-1)e^{-r_{deg}t} + 1\right]^{n_0} e^{I_t(z-1)} \qquad \text{with } I_t = \int_0^t R_\tau e^{-\int_\tau^t r_{deg} d\tau'} d\tau. \qquad (4.3)$$

Again, the first factor of $G(z,t|n_0)$ reflects the dependence of the distribution on the initial value n_0. The second factor $\exp(I_t(z-1))$ corresponds to the long-term behavior of the mRNA content and equals the time-dependent probability generating function of a Poisson distribution with intensity parameter I_t (see Definition 3.3) that is now generalized as some stochastic process. Analogously, in steady state the first part vanishes so that only the second part remains and describes the mRNA count distribution. Again, the mRNA counts follow a compound Poisson distribution with intensity parameter I_t being governed by the transcription and degradation process. From Definition 3.3, we get

$$\mathcal{P}_{\text{steady state}}(n,t) = \mathcal{P}_{I_t}(n,t) = \int_0^\infty \frac{x^n}{n!} e^{-x} f_{I_t}(x,t)dx \tag{4.4}$$

for $n \in \mathbb{N}_0$ and $t \geq 0$ (but large), where f_{I_t} denotes the density of I_t. Remember that the intensity function in the basic model could be further simplified. To exactly specify the compound Poisson distribution we need to take a closer look at the intensity process I_t, defined through (4.3), and examine its long-term (steady state) behavior. $I_t = \int_0^t R_\tau e^{-\int_\tau^t r_{deg} d\tau'} d\tau$ is a solution of the random differential equation (RDE)

$$\frac{\mathrm{d}I_t}{\mathrm{d}t} + r_{deg} I_t = R_t,$$

which can be rewritten as

$$\mathrm{d}I_t = -r_{deg} I_t \mathrm{d}t + R_t \mathrm{d}t, \tag{4.5}$$

for $t \geq 0$ and fixed $I_0 = i_0 > 0$. In this representation, we can directly recognize the impact of the mRNA degradation rate r_{deg} and the transcription rate R_t on the number of mRNA molecules: Larger r_{deg} will lead to lower mRNA numbers, larger R_t to higher numbers. The properties and steady state of I_t clearly depend on the choice of R_t. Depending on the transcription process R_t this RDE has different solutions.

Example 4.1 (Deterministic Continuous Transcription Model) *If R_t is a deterministic rather than a stochastic function $R(t)$, I_t itself becomes deterministic, then denoted by $I(t)$. Dattani and Barahona (2017) show that the probability to have n mRNA molecules at time t is a Poisson distribution with time-dependent intensity $I(t)$, i. e.*

$$\mathcal{P}_{\text{Pois}-I(t)}(n,t) = \frac{I(t)^n}{n!} e^{-I(t)}.$$

The solution for $I(t)$ is then analogous to the calculations in Appendix D.1 given by

$$I(t) = \int_0^t R(\tau) e^{-\int_\tau^t r_{\text{deg}} d\tau'} d\tau = \int_0^t R(\tau) e^{-r_{\text{deg}}(t-\tau)} d\tau = e^{-r_{\text{deg}} t} \int_0^t R(\tau) e^{r_{\text{deg}} \tau} d\tau. \tag{4.6}$$

Example 4.2 (Basic Transcription Model) *The basic model in Section 4.2.1 equals the special case of the deterministic continuous transcription model in Example 4.1, when $R(t)$ takes only one time-independent value r_{tran}. The RDE* (4.5) *then simplifies to the ordinary differential equation (ODE)*

$$\mathrm{d}I(t) = -r_{\text{deg}} I(t)\mathrm{d}t + r_{\text{tran}}\mathrm{d}t, \tag{4.7}$$

and Equation (4.6) *simplifies to*

$$I(t) = r_{\text{tran}} e^{-r_{\text{deg}} t} \left(\frac{e^{r_{\text{deg}} t} - 1}{r_{\text{deg}}} \right) = \frac{r_{\text{tran}}}{r_{\text{deg}}} \left(1 - e^{-r_{\text{deg}} t}\right).$$

All together, for $t \to \infty$, the steady state distribution of the mRNA counts follows a Poisson distribution with constant intensity parameter $I = r_{\text{tran}}/r_{\text{deg}}$.

4.2.3 Switching Model: Gene Activation and Deactivation

In the well-known switching model, a gene switches between an inactive state where transcription is impossible, and an active state where transcription occurs. This can be explained by polymerases binding and unbinding to the specific gene as depicted in Figure 4.3A. Transcription is assumed to be governed by $R_t = r_{switch}(t)$, which is a continuous-time Markov process (see Definition 3.7) with two states *on* (or *active*) and *off* (or *inactive*). A Switching between these two states happens after exponentially distributed waiting times with rates r_{act} and r_{deact}. During the *active* state, transcription happens with rate r_{on}, whereas in the *inactive* state, either strongly down-regulated transcription happens (small r_{off}) or none ($r_{off} = 0$). Figure 4.3B depicts this detailed process.

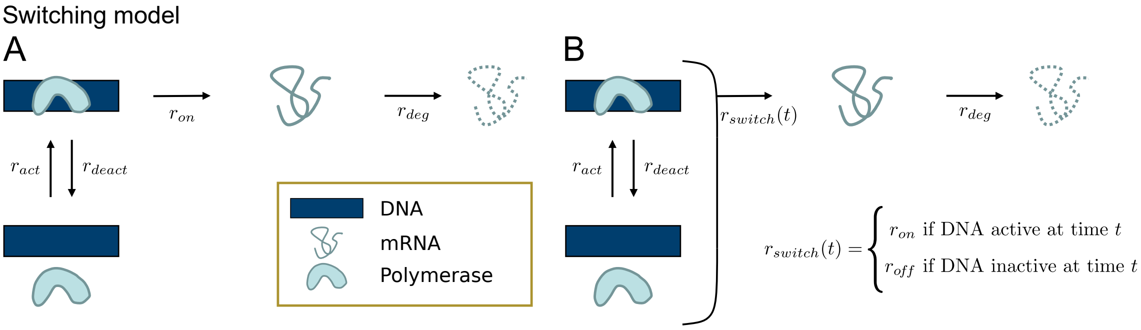

Switching model
A
r_{on}
r_{deg}
r_{act}
r_{deact}
DNA
mRNA
Polymerase
B
$r_{switch}(t)$
r_{deg}
r_{act}
r_{deact}
$r_{switch}(t) = \begin{cases} r_{on} & \text{if DNA active at time } t \\ r_{off} & \text{if DNA inactive at time } t \end{cases}$

Figure 4.3: (A) Switching model as model of gene activation and inactivation, transcription and degradation. (B) Switching model as model of transcription – modeled by a two-state Markov process – and degradation.

The steady state distribution of mRNA content can be calculated by following the derivation of Smiley and Proulx (2010), who show how to obtain the density function for the mRNA expression level. Dattani and Barahona (2017) use this result and transfer it into the probability distribution. Raj et al. (2006) arrive at the same solution. The complete calculation in our notation can be found in Appendix D.3. The RDE (4.5) becomes

$$dI_t = -r_{deg} I_t \mathrm{d}t + r_{switch}(t)\mathrm{d}t \tag{4.8}$$

with

$$r_{switch}(t) = \begin{cases} r_{on} & \text{if DNA active at time } t \\ r_{off} & \text{if DNA inactive at time } t, \end{cases}$$

where $r_{off} < r_{on}$. The transcription rate is modeled by a continuous-time Markov process $(r_{switch}(t))_{t\geq 0}$ that switches between two discrete states r_{on} and r_{off} with activation and deactivation rates r_{act} and r_{deact}, respectively. Again, the probability distribution for the amount of mRNA at time t is a compound Poisson distribution as described by Equation (4.4). To determine the steady state distribution of mRNA counts, the steady state distribution of I_t governed by $(r_{switch}(t))_{t\geq 0}$ in Equation (4.8) needs to be calculated. Following the calculations in Appendix D.3, we find in Equation (D.16) that I_t follows in steady state a four-parametric beta distribution (see Definition A.5) with parameters $a = r_{off}/r_{deg}$, $c = r_{on}/r_{deg}$, $\alpha = r_{act}/r_{deg}$ and $\beta = r_{deact}/r_{deg}$.
The overall steady state distribution of mRNA counts is by construction a compound Poisson distribution (see Equation (4.4)). When conditioning the Poisson distribution on an intensity parameter following a beta distribution defined by Equation (D.16), the overall distribution will be a (compound) PB distribution with PMF given in Example 3.4. Taken together, the probability of having n mRNA molecules at time t is time-independent. With the parameters given above and for $r_{off} = 0$ (i. e. no transcription possible during inactive DNA state), the PMF can be simplified as shown in Equation (3.1) to

$$\mathcal{P}(n,t) = \frac{\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{on}}{r_{deg}}\right)^n \Gamma\left(\frac{r_{act}}{r_{deg}} + n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}} + n\right)} \; {}_1F_1\left(\frac{r_{act}}{r_{deg}}+n; \frac{r_{deact}}{r_{deg}}+\frac{r_{act}}{r_{deg}}+n; -\frac{r_{on}}{r_{deg}}\right), \tag{4.9}$$

where ${}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^1 e^{zu}u^{a-1}(1-u)^{b-a-1}du$ is the confluent hypergeometric function of first order, also called Kummer function (see Definition A.6) and Γ denotes the gamma function.

4.2.3.1 Asymptotic Limits of the Switching Model

The density function of this PB $(r_{act}/r_{deg}, r_{deact}/r_{deg}, 0, r_{on}/r_{deg})$ distribution converges to the density function of a NB distribution under specific conditions. For large r_{deact}/r_{deg} and $r_{on}/r_{deact} < 1$, the PMF of this distribution converges towards the one of a NB distribution (see Definition A.8 and Raj et al., 2006):

$$\begin{aligned}
&\mathcal{P}_{\mathrm{PB}\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}, 0, \frac{r_{on}}{r_{deg}}\right)}(X=n) \\
&= \frac{\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{on}}{r_{deg}}\right)^n \Gamma\left(\frac{r_{act}}{r_{deg}} + n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}} + n\right)} \; {}_1F_1\left(\frac{r_{act}}{r_{deg}}+n; \frac{r_{deact}}{r_{deg}}+\frac{r_{act}}{r_{deg}}+n; -\frac{r_{on}}{r_{deg}}\right) \\
&= \frac{\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{on}}{r_{deg}}\right)^n \Gamma\left(\frac{r_{act}}{r_{deg}} + n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)\Gamma\left(\frac{r_{act}}{r_{deg}} + \frac{r_{deact}}{r_{deg}} + n\right)}
\end{aligned}$$

$$\sum_{l=0}^{\infty}\left[\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n+l\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n+l\right)}\frac{\left(-\frac{r_{on}}{r_{deg}}\right)^{l}}{l!}\right]$$

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)\left(\frac{r_{on}}{r_{deg}}\right)^{n}}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\sum_{l=0}^{\infty}\left[\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n+l\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n+l\right)}\frac{\left(-\frac{r_{on}}{r_{deg}}\right)^{l}}{l!}\right]$$

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)\left(\frac{r_{on}}{r_{deg}}\right)^{n}}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}$$

$$\sum_{l=0}^{\infty}\left[\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n+l\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)\Gamma(l+1)}\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n+l\right)}\left(\frac{r_{deact}}{r_{deg}}\right)^{l}\left(-\frac{r_{on}}{r_{deact}}\right)^{l}\right]$$

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)\left(\frac{r_{on}}{r_{deg}}\right)^{n}}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\sum_{l=0}^{\infty}\left[\binom{\frac{r_{act}}{r_{deg}}+n+l-1}{\frac{r_{act}}{r_{deg}}+n-1}\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}\right)}{\Gamma\left(\frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{deact}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}}}\right.$$

$$\left.\cdot\frac{\Gamma\left(\frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{deact}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}+n+l}}{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n+l\right)}\left(\frac{r_{deact}}{r_{deg}}\right)^{-n-l}\left(\frac{r_{deact}}{r_{deg}}\right)^{l}\left(-\frac{r_{on}}{r_{deact}}\right)^{l}\right]$$

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\left(\frac{r_{on}}{r_{deg}}\right)^{n}\left(\frac{r_{deg}}{r_{deact}}\right)^{n}\sum_{l=0}^{\infty}\left[\binom{\frac{r_{act}}{r_{deg}}+n+l-1}{\frac{r_{act}}{r_{deg}}+n-1}\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}\right)}{\Gamma\left(\frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{deact}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}}}\right.$$

$$\left.\cdot\frac{\Gamma\left(\frac{r_{deact}}{r_{deg}}\right)\left(\frac{r_{deact}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}+n+l}}{\Gamma\left(\frac{r_{act}}{r_{deg}}+\frac{r_{deact}}{r_{deg}}+n+l\right)}\left(-\frac{r_{on}}{r_{deact}}\right)^{l}\right].$$

Taking the limit of $\frac{r_{deact}}{r_{deg}}$ to infinity, the asymptotic approximation given in Identity 1 can be applied twice. Therefore,

$$\lim_{\frac{r_{deact}}{r_{deg}}\to\infty}\mathcal{P}_{\mathrm{PB}\left(\frac{r_{act}}{r_{deg}},\frac{r_{deact}}{r_{deg}},0,\frac{r_{on}}{r_{deg}}\right)}(X=n)$$

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\left(\frac{r_{on}}{r_{deact}}\right)^{n}\sum_{l=0}^{\infty}\left[\binom{\frac{r_{act}}{r_{deg}}+n+l-1}{\frac{r_{act}}{r_{deg}}+n-1}\left(-\frac{r_{on}}{r_{deact}}\right)^{l}\right].$$

Using Equation (B.1) with $r_{on}/r_{deact}<1$ simplifies to:

$$=\frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\left(\frac{r_{on}}{r_{deact}}\right)^{n}\frac{1}{\left(1+\frac{r_{on}}{r_{deact}}\right)^{\frac{r_{act}}{r_{deg}}+n}}$$

$$
\begin{aligned}
&= \frac{\Gamma\left(\frac{r_{act}}{r_{deg}}+n\right)}{\Gamma\left(\frac{r_{act}}{r_{deg}}\right)\Gamma(n+1)}\left(\frac{\frac{r_{on}}{r_{deact}}}{1+\frac{r_{on}}{r_{deact}}}\right)^{n}\left(1+\frac{r_{on}}{r_{deact}}\right)^{-\frac{r_{act}}{r_{deg}}}\\
&= \binom{\frac{r_{act}}{r_{deg}}+n-1}{n}\left(1-\frac{r_{deact}}{r_{deact}+r_{on}}\right)^{n}\left(\frac{r_{deact}}{r_{deact}+r_{on}}\right)^{\frac{r_{act}}{r_{deg}}}\\
&= \mathcal{P}_{\mathrm{NB}\left(\frac{r_{act}}{r_{deg}},\frac{r_{deact}}{r_{deact}+r_{on}}\right)}(X=n).
\end{aligned}
$$

This is the PMF of the $\mathrm{NB}\left(r_{act}/r_{deg}, r_{deact}/(r_{deact}+r_{on})\right)$ distribution.

Taken together, for $r_{on} = r_{tran}$, $r_{act} \to \infty$ and $r_{deact} = 0$, the switching model reduces to the basic model, and the above PB distribution collapses to a Poisson distribution with intensity parameter r_{tran}/r_{deg}, in consistency with the above-derived results.

4.2.3.2 Connecting to Queuing Systems

The switching model can also be transformed into the following queuing system: An officer puts up a sign: “Please do not queue anymore”, i. e. queuing is not possible for some time. The system is unchanged w.r.t. rates compared to the basic model, but additional rates regulate the putting up and taking away of the sign. Setting up the sign is done with rate r_{deact} and is only possible if there is no sign at the moment. Removing the sign is done at rate r_{act} and is only possible if the sign is currently there. As long as the sign is present, no new customers can arrive, but present customers can still leave as depicted in Figure 4.4. For this complex scenario no example is present in Adan and Resing (2015).

4.3 Connecting SDEs with Steady State Distributions

Both models – the basic and the switching model – lead to mRNA counts which follow Poisson distributions with different intensity processes, see Equations (4.7) and (4.8). These intensity processes are governed by the respective transcription and degradation mechanisms. They determine the steady state distribution of the intensity parameter, and thus the overall steady state distribution of the mRNA content. Importantly, changes in the intensity process lead to different steady state distributions.

Next, we generalize the RDE (4.5) to a stochastic differential equation by considering $R_t\mathrm{d}t = \mathrm{d}L_t$, where L_t is an arbitrary increasing Lévy process (also called subordinator, see Definition 3.9). Then

$$\mathrm{d}I_t = -r_{deg}I_t\mathrm{d}t + \mathrm{d}L_t. \tag{4.10}$$

for $t \geq 0$ and fixed $I_0 = i_0 > 0$. Since the trajectories of a Lévy process are not necessarily left-continuous, their derivatives may not exist in the classical sense. Care has to be taken here. In the following, we show how to derive the steady state distribution of I_t for different choices of L_t.

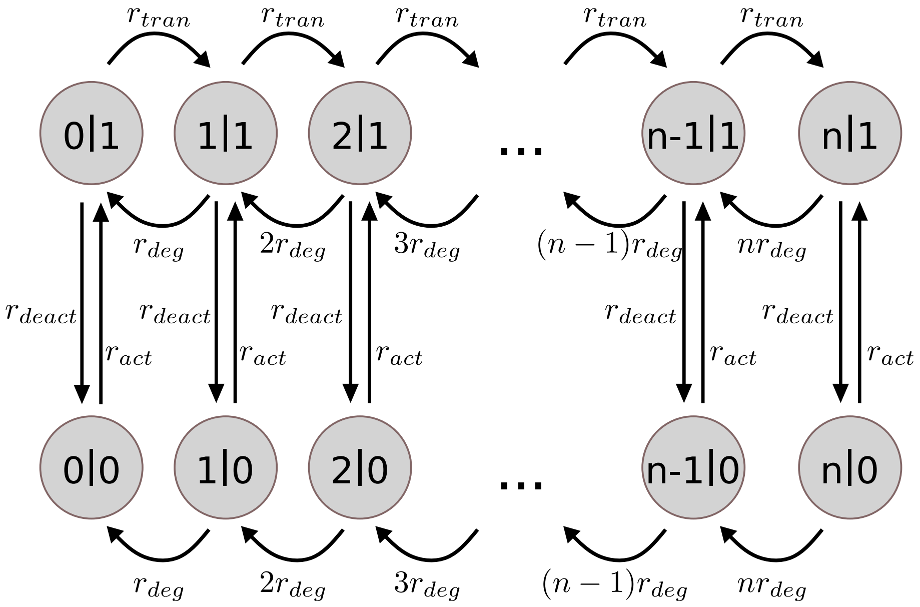


Figure 4.4: Switching model with all possible states and transitions: First component shows the number of mRNA, the count we want to model via the queuing theory, the second component the state of the DNA: which is either "on" or "off".

The generalized process given in Equation (4.10) with subordinator L_t can be identified as an Ornstein-Uhlenbeck (OU) process, see Definition 3.5. OU processes and the concept of linking them to distributions is widely used in financial mathematics, especially in the areas of option pricing and volatility modeling. Among others (Rogers and Williams, 2000, Sato, 1999), especially Barndorff-Nielsen and Shephard (2001b) and Barndorff-Nielsen and Shephard (2001a) used OU processes in a wide range and showed and proved a substantial amount of their properties. In the following we will use properties of OU processes and apply them to the generalized intensity process in (4.10).

Lemma 4.1 (Barndorff-Nielsen and Shephard, 2001a) *A special property of OU processes is that, given a one-dimensional distribution $\mathcal{D}$, there exists an OU–type stationary process whose one-dimensional law is $\mathcal{D}$ if and only if $\mathcal{D}$ is self-decomposable (see Definition 3.6).*

This means that, under these specific conditions for a chosen distribution $\mathcal{D}$ there is an OU process that in steady state leads to this distribution $\mathcal{D}$. The other direction, i. e. the existence of a steady state distribution $\mathcal{D}$ for a chosen OU process (in terms of its subordinator), holds as well. In most applications in financial mathematics, the SDE (3.5) is transformed to

$$\mathrm{d}y_t = -\lambda y_t \,\mathrm{d}t + \mathrm{d}z_{\lambda t} \qquad \text{for some } \lambda > 0$$

such that whatever value of λ is chosen, the marginal distribution of y_t remains unchanged. However, in the context of this thesis, we work with the original, untransformed SDE (4.10). Hence, in our notation, for a given Lévy subordinator L_t, the characteristic function of $\mathcal{D}$, and thus $\mathcal{D}$ itself, can be derived as follows (Barndorff-Nielsen et al., 1998, Sato, 1999):

1. Find the characteristic function $\hat{\mu}_{L_t}(z)$ of the Lévy subordinator L_t.
2. Calculate $\hat{\mu}_{L_1}(z)$ and write the result in the form $\exp(\phi(z))$ for some function $\phi(z)$.
3. Calculate the characteristic function $\hat{\mu}_{I_t}(z)$ of the stationary distribution $\mathcal{D}$ of I_t by setting $\hat{\mu}_{I_t}(z) = \exp(r_{deg}^{-1} \int_0^z \phi(\omega)\omega^{-1}\,\mathrm{d}\omega)$. Identifying $\hat{\mu}_{I_t}(z)$ leads to $\mathcal{D}$.

Despite this apparently clear line of action, finding a corresponding law $\mathcal{D}$ and process L_t is challenging without prior knowledge, e. g. if $\mathcal{D}$ is not well-known or L_t is only specified through the characteristic function of L_1.
As an example we will show the derivation of the steady state distribution of the basic model.

Example 4.3 (OU Process Derivation for the Basic Model) *In the following, we will show how to use an OU process and its properties to infer the steady state distribution of the basic model shown in Figure 4.1. We use the following general OU equation introduced in Equation* (4.10)*:*

$$\mathrm{d}I_t = -r_{\mathrm{deg}} I_t\,\mathrm{d}t + \mathrm{d}L_t.$$

This general SDE is transformed to the ODE of the basic model by setting $L_t := r_{\mathrm{tran}}t$, with $\mathrm{d}L_t = r_{\mathrm{tran}}\mathrm{d}t$, yielding the ODE

$$\mathrm{d}I_t = -r_{\mathrm{deg}} I_t\,\mathrm{d}t + r_{\mathrm{tran}}\mathrm{d}t,$$

which was already given in Equation (4.7)*. In this simple case, the Lévy subordinator $L_t = r_{\mathrm{tran}}t$ describes a state-continuous process without any jumps or Brownian components. Still, this ODE fulfills all required properties and can be used for deriving a steady state distribution for the mechanistic model according to the procedure that was described before.*
To do so, we now follow the three steps described above:

1. *Find the characteristic function of the Lévy subordinator $L_t = r_{\mathrm{tran}}t$. For the basic model, that is*

$$\hat{\mu}_{L_t}(z) = \mathbf{E}[\exp(izr_{\mathrm{tran}}t)] = \exp(izr_{\mathrm{tran}}t).$$

2. *Calculate $\hat{\mu}_{L_1}(z)$ and write the result in the form* $\exp(\phi(z))$ *to determine $\phi(z)$. For the basic model, that is*

$$\hat{\mu}_{L_1}(z) = \exp(\underbrace{izr_{\mathrm{tran}}}_{=\phi(z)}),$$

so it follows that $\phi(z) = izr_{\mathrm{tran}}$.

3. *Calculate the characteristic function $\hat{\mu}_{I_t}(z)$ of the stationary distribution $\mathcal{D}$ of I_t by*

$$\hat{\mu}_{I_t}(z) = \exp\left(r_{\text{deg}}^{-1} \int_0^z i\omega r_{\text{tran}} \omega^{-1} \, \mathrm{d}\omega \right) = \exp\left(\frac{i r_{\text{tran}} z}{r_{\text{deg}}} \right).$$

This is the characteristic distribution of a point distribution where all mass is concentrated at a single point $r_{\text{tran}}/r_{\text{deg}}$,see Example 2.19 in Sato (1999). This equals the solution that we obtained in Section 4.2.1 by solving the master equation directly.

In the following, we cast the NB distribution as an alternative distribution for which a subordinator can be derived.

4.3.1 Negative Binomial Distribution: Deriving an Explanatory Bursting Process

A widely considered model for scRNA-seq counts is the NB distribution. Like the PB distribution – the steady state distribution of the switching model as shown in Section 4.2.3 –, it accounts for overdispersion by modeling the variance independently of the mean of the data. Having one parameter less than the PB, the NB distribution is an appealing choice. However, mechanistic models underlying the NB distributional assumption for the steady state distribution of mRNA content have not been formulated explicitly before. We aim to derive such a mechanistic model of transcription and degradation by reversing the steps that led from the switching model to the PB distribution. For that purpose, an important fact is that a NB distribution can be expressed as a Poisson-gamma (PG) distribution , i. e. as a conditional Poisson distribution with gamma distributed intensity parameter I. In Example 3.3, we showed that

$$\text{PG}(\alpha, \beta) \mathrel{\hat{=}} \text{NB}\left(\alpha, \frac{\beta}{\beta+1}\right) \tag{4.11}$$

for $\alpha, \beta > 0$.

In analogy to the derivation of the PB distribution from the switching model, we now seek to describe the mRNA content by a Poisson distribution with intensity parameter I_t, which in steady state follows a gamma distribution instead of a beta distribution.

Thus, we aim to specify an OU process (4.10) with the gamma distribution as steady state distribution. In terms of mechanistic modeling, this means that we need to describe a suitable transcription process. Mathematically, we need to specify the Lévy subordinator L_t accordingly. But it is important to remember that this is only the first step in finding the underlying transcriptional mechanism. In case the subordinator is found, only the intensity process is inferred. This completed intensity process can then be used to possibly infer the underlying transcriptional mechanism.

4.3.1.1 Deriving the Subordinator

From financial mathematics it is known that a stationary gamma distribution is obtained if L_t is chosen to be a compound Poisson process (CPP, see Definition 3.11) with exponentially distributed jump sizes (Barndorff-Nielsen and Shephard, 2001a). This will be our choice of subordinator; however, the parameters of this process still need to be specified.

In the following, we will show that the Lévy subordinator of the Ornstein-Uhlenbeck process (4.10) whose one-dimensional stationary distribution is $\text{Gamma}(\alpha, \beta)$, is a compound Poisson process (CPP) with intensity parameter $\alpha \cdot r_{deg}$ and mean jump size β^{-1}.

To obtain this result, we follow the three-step procedure described above in reverse order. We start with $\mathcal{D} \hat{=} \text{Gamma}(\alpha, \beta)$ and transform its characteristic function to $\exp\left\{r_{deg}^{-1} \int_0^z \phi(\omega)\omega^{-1} d\omega\right\}$, using the characteristic function of $\mathcal{D}$ as given in Definition A.3:

$$\begin{aligned}
\hat{\mu}_{I_t}(z) = \left(1 - \frac{iz}{\beta}\right)^{-\alpha} &= \exp\left\{-\alpha \log\left(1 - \frac{iz}{\beta}\right)\right\} \\
&= \exp\left\{\alpha \int_0^z \frac{-1}{i\beta + \omega} \mathrm{d}\omega\right\} \\
&= \exp\left\{\alpha \int_0^z \frac{i\omega}{(\beta - i\omega)\omega} \mathrm{d}\omega\right\} \\
&= \exp\left\{r_{deg}^{-1} \int_0^z \alpha\, r_{deg} \left(\frac{\beta}{\beta - i\omega} - 1\right) \omega^{-1} \mathrm{d}\omega\right\} \\
&= \exp\left\{r_{deg}^{-1} \int_0^z \phi(\omega)\omega^{-1} d\omega\right\}
\end{aligned}$$

with $\phi(\omega) = \alpha\, r_{deg} \left(\frac{\beta}{\beta - i\omega} - 1\right)$ and i the imaginary number. Next, we use $\hat{\mu}_{L_1}(z) = \exp(\phi(z))$ to obtain

$$\hat{\mu}_{L_1}(z) = \exp\left(\alpha\, r_{deg} \left(\frac{\beta}{\beta - iz} - 1\right)\right). \tag{4.12}$$

We aim to bring this into agreement with $\hat{\mu}_{L_t}(z)$, the time-dependent characteristic function of a general compound Poisson process L_t with intensity parameter λ. This is given by

$$\hat{\mu}_{L_t}(z) = \exp(t\, \lambda(\hat{\mu}_Y(z) - 1)),$$

where Y is a random variable following the distribution of the jump sizes of the compound Poisson process, and $\hat{\mu}_Y$ is its characteristic function (see Definition 3.11). A compound Poisson process with intensity $\lambda = \alpha \cdot r_{deg}$ and i.i.d. exponentially distributed increments $Y \sim \text{EXP}(\beta)$ with characteristic function $\hat{\mu}_Y(z) = \beta/(\beta - iz)$ yields the overall characteristic function

$$\hat{\mu}_{L_t}(z) = \exp\left(t\alpha r_{deg} \left(\frac{\beta}{\beta - iz} - 1\right)\right).$$

This is in accordance with $\hat{\mu}_{L_1}(z)$ as derived in Equation (4.12), and hence, a mathematically appropriate subordinator is a compound Poisson process with intensity parameter $\alpha \cdot r_{deg}$ and mean jump size β^{-1}.

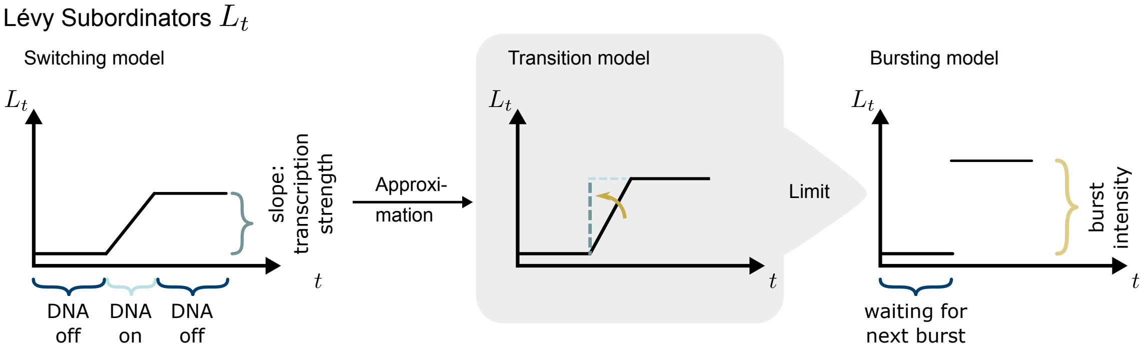


Figure 4.5: The Lévy subordinator of the switching model is shown on the left by means of an exemplary trajectory. As depicted in the middle (transition model), this trajectory moves (yellow arrow) up as soon as the duration of the DNA being active gets smaller accompanied by a larger transcription strength. The limit of this approximation, with infinitesimally small DNA activation time interval and infinitesimally large transcription strength, leads to a trajectory of the subordinator of the bursting model that is now described by a step function which is shown on the right.

4.3.1.2 Deriving the Underlying Mechanism

As a consequence, transcription is expressed via a stochastic process L_t, namely the CPP, which experiences jumps after exponentially distributed waiting times. In contrast to the Lévy subordinators of the basic model, $L_t^{\text{basic}} = r_{on}t$, and of the switching model, $L_t^{\text{switch}} = \int_0^t r_{\text{switch}}(s)ds$, it possesses pointwise discontinuous sample paths (Figure 4.5, right). Intervals without any transcription activity seem to be disrupted by sudden explosions of mRNA numbers. This burstiness led us to call the mechanism behind the NB distribution the bursting model. We denote its subordinator by L_t^{burst} and argue the biological justification of the model in the Section 4.6.

We aim to derive the mechanistic transcription process of the bursting model in more detail. Specifically, we tackle the distribution of burst sizes of mRNA counts. For this we look at a heuristic transition from L_t^{switch} to L_t^{burst}.

First, we dismantle the shape of the trajectories of L_t^{switch}. As depicted in Figure 4.5 on the left, such a trajectory consists of alternating piecewise constant and piecewise linear parts. The constant parts appear during time intervals without transcription, i. e. where the DNA is inactive. The length of such a time interval depends only on the rate r_{act} of the switching model. Once the DNA switches into the active mRNA transcribing state, the time interval with transcription depends only on the rate r_{deact}. The slope of the trajectory during this active DNA state represents the transcription strength and depends only on the parameter r_{on}.

In case the length of the time interval of active DNA becomes infinitesimally small, and at the same time the transcription strength becomes infinitesimally large, the trajectory of L_t^{switch} turns into a step function as depicted in Figure 4.5 on the right. This limit is obtained if $r_{deact} \to \infty$ and $r_{on} \to \infty$ in a way that needs to be specified. For that reason, we in the following seek to describe a mechanistic model for the transition phase (Figure 4.5, middle) leading to the bursting model.

In the switching model, as soon as DNA becomes active, a competition starts between the events *transcription* and *deactivation*. In addition, degradation may happen, which will affect the intensity process I_t and the number of mRNA molecules, but not the transcription process. If a transcription event occurs, the competition between transcription and deactivation continues at the same probability rates as before; the only affected event probability is the one for degradation because this probability depends on the current mRNA count. We now consider the following approximation of the switching model and call it the transition model: When DNA becomes active, we allow the events transcription and deactivation to happen, but not degradation. To correct for the missing degradation events, we introduce a waiting time W after DNA deactivation in which only degradation can occur, but no DNA activation. For appropriately chosen $r_{deact} \to \infty$ and $r_{on} \to \infty$, the approximation error will tend to zero.

The number of transcription events S during one active DNA phase is geometrically distributed with success probability parameter $r_{deact}/(r_{deact} + r_{on})$. In the interpretation of the geometric distribution, transcription events are considered as failures, deactivation as success. The waiting time W needs to accumulate the times before S transcriptions and one deactivation. Thus, $W = T_1 + \cdots + T_S + D$, where $T_i \sim \text{EXP}(r_{on}),\ i = 1, \ldots, S$, are the single waiting times for each transcription event and $D \sim \text{EXP}(r_{deact})$ is the waiting time till the next DNA deactivation.

Taken together, the bursting process can be considered as the limiting process of the approximation of the switching process as $r_{on} \to \infty$ and $r_{deact} \to \infty$ under the condition that the success probability parameter of the geometric distribution, $r_{deact}/(r_{deact} + r_{on})$ stays constant. As the link between the switching model and the PB distribution is known, and since a PB distribution converges towards a NB distribution under certain conditions (Section 4.2.3.1), we can connect the parameters of the bursting model with those of the NB distribution and the compound Poisson process.

That is, the bursting model can mechanistically be described as follows: After $\text{EXP}(r_{burst})$-distributed waiting times, a $\text{Geo}((1 + s_{burst})^{-1})$-distributed number of mRNAs are produced at once, where s_{burst} is the mean burst size (see also Golding et al., 2005). As in the basic and switching models, degradation events occur with $\text{EXP}(r_{deg})$-distributed waiting times.

The just described mechanistic bursting model is shown in Figure 4.6. It can equivalently be described by the OU process (4.10) with L_t being a compound Poisson process with $\text{EXP}(r_{burst})$-distributed waiting times and $\text{EXP}(s_{burst}^{-1})$-distributed jump sizes. Thus, in steady state, mRNA counts follow a $\text{PG}(r_{burst}/r_{deg}, {s_{burst}}^{-1})$ distribution or, equivalently with Equation (4.11), a $\text{NB}(r_{burst}/r_{deg}, (1 + s_{burst})^{-1})$ distribution if the bursting model is assumed.

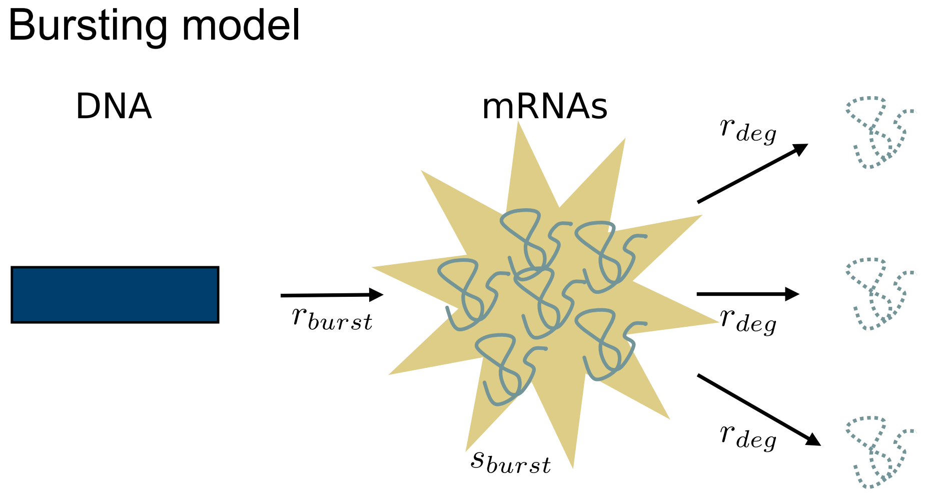


Figure 4.6: Bursting Model of gene expression consists of transcription, where bursts create geometrically distributed bulks of mRNA and their constant degradation.

The NB$(r_{burst}/r_{deg}, (1+s_{burst})^{-1})$ model, again, can be interpreted as follows: Assume you have an empty bucket into which you put balls according to the following stochastic procedure. You perform a number of independent Bernoulli trials, each with success probability $(1+s_{burst})^{-1}$. If there is a failure, you add one ball to the bucket. If there is a success, you do not do anything but count the success event. You continue until there have been r_{burst}/r_{deg} successes. (For interpretation purposes, we here assume r_{burst}/r_{deg} to be a whole-valued number.) The larger s_{burst}, the smaller the success probability, i. e. by expectation you will put more balls in the bucket for large s_{burst}. Similarly, the larger the ratio of r_{burst} to r_{deg}, the more success events will be waited for, thus the more balls will tend to be added. The final number of balls in the bucket represents the number of mRNA molecules in a cell at steady state.

The above top-down derivation from the steady state distribution to the mechanistic process is motivated heuristically in parts. Next, we prove bottom-up that the above described mechanistic bursting model indeed leads to the steady state NB distribution by directly calculating the master equation. To do so we first set up the corresponding queuing system

4.3.1.3 Queuing System and Master Equation

When the mechanistic model of the bursting process in known, its master equation can be set up easily, especially if one draws a connection to queuing theory. In a general queuing model, customers arrive at one or several service desks according to some arrival process, which in our case corresponds to the transcription process. The number of customers waiting is equivalent to the number of mRNA molecules in a cell. As soon as a customer can proceed from the queue to a service desk, this number decreases by one, corresponding to mRNA degradation. Here, service time is zero and thus plays no role in our model.

The bursting model described in the main text corresponds to the following queuing

system: Customers do not arrive separately at constant rate, but they arrive in groups (e. g., in buses) after exponentially distributed waiting times with rate r_{burst}. Then, several people start queuing at the same time. The number of people arriving with each group follows a geometric distribution with mean s_{burst}.
This process corresponds to a mixture of two queuing problems from Adan and Resing (2015). The first queuing problem is the basic $M/M/\infty$ queuing setup (Example 11.1.1 in that reference), and the second one is the $M/G/1$ model which corresponds to a queue with group arrivals (Chapter 10.4 in that reference). (The notation here is due to Kendall: In the three-part code $a/b/c$, a specifies the inter-arrival time distribution, b the service time distribution and c the number of servers. The letter G is used for a general distribution, M for the exponential distribution and D for deterministic times.) A standard waiting process is modeled where the group arrival time is exponentially distributed, service time and group size follow arbitrary distributions, but only one service counter is open. With those two models in mind, we set up our bursting queuing process (as mentioned above we do not have service times). We illustrate all possible state changes in Figure 4.7.

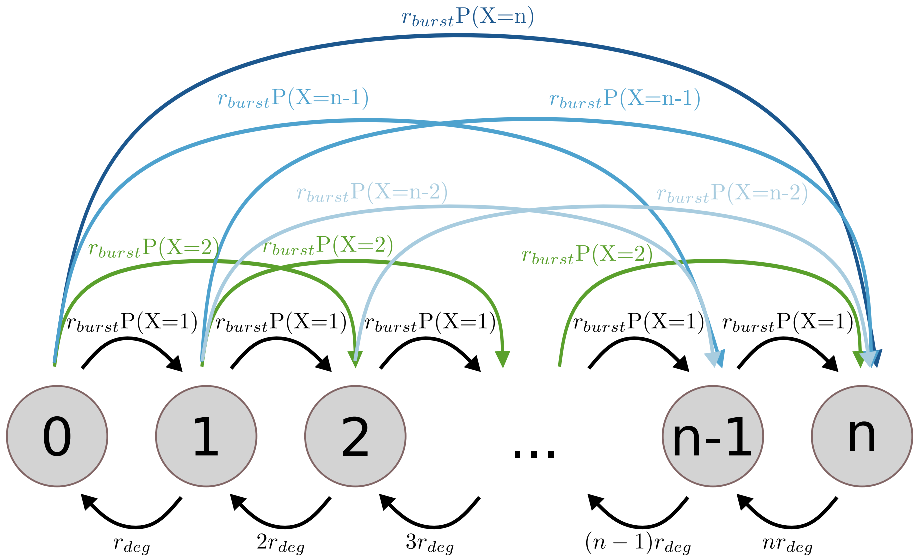


Figure 4.7: Bursting model with all states and possible transitions between states, assuming that at most one event (transcription or degradation) can happen at the same time. Transitions from one node to itself are not depicted. Here, $P(X = k)$ stands for the probability of a geometrically distributed random variable X taking the value k.

Along Figure 4.7, we can set up the master equation directly:

$$\begin{aligned}\frac{d\mathcal{P}(n,t)}{dt} &= \sum_{x=0}^{\infty} r_{burst}\mathcal{P}(n-x,t)\mathrm{P}_{\mathrm{Geo}}(X=x) + r_{deg}(n+1)\mathcal{P}(n+1,t)\\ &\quad - \left(\sum_{x=0}^{\infty} r_{burst}\mathrm{P}_{\mathrm{Geo}}(X=x) + r_{deg}\,n\right)\mathcal{P}(n,t),\end{aligned}$$

where $\mathrm{P}_{\mathrm{Geo}}$ denotes the PMF of a random variable X that is geometrically distributed with success probability p (see Definition A.9). The derivative of the probability-generating function then reads

$$\begin{aligned}\frac{\partial G}{\partial t}(z,t) =&\sum_{n=0}^{\infty} z^n \frac{d\mathcal{P}(n,t)}{dt}\\ =&\sum_{x=0}^{\infty} z^x r_{burst}\mathrm{P}_{\mathrm{Geo}}(X=x)\sum_{n=0}^{\infty} z^{n-x}\mathcal{P}(n-x,t) + r_{deg}\sum_{n=0}^{\infty}(n+1)z^n\mathcal{P}(n+1,t)\\ &- r_{burst}\sum_{x=0}^{\infty}\mathrm{P}_{\mathrm{Geo}}(X=x)\sum_{n=0}^{\infty} z^n\mathcal{P}(n,t) - r_{deg}z\sum_{n=0}^{\infty} nz^{n-1}\mathcal{P}(n,t).\end{aligned}$$

With

$$\begin{aligned}G(z,t) &= \sum_{n=0}^{\infty} z^{n-x}\mathcal{P}(n-x,t)\\ G(z,t) &= \sum_{n=0}^{\infty} z^{n}\mathcal{P}(n,t)\\ \frac{\partial G}{\partial z}(z,t) &= \sum_{n=0}^{\infty} nz^{n-1}\mathcal{P}(n,t)\\ \frac{\partial G}{\partial z}(z,t) &= \sum_{n=0}^{\infty}(n+1)z^{n}\mathcal{P}(n+1,t)\end{aligned}$$

it follows that

$$\frac{\partial G}{\partial t}(z,t) = r_{burst}\left(\sum_{x=0}^{\infty} z^x\mathrm{P}_{\mathrm{Geo}}(X=x) - \sum_{x=0}^{\infty}\mathrm{P}_{\mathrm{Geo}}(X=x)\right)G(z,t) + r_{deg}(1-z)\frac{\partial G}{\partial z}(z,t).$$

Because of $\mathrm{P}_{\mathrm{Geo}}(X=x) = (1-p)^x p$ and $\sum_{x=0}^{\infty}\mathrm{P}_{\mathrm{Geo}}(X=x) = 1$, it follows that

$$\frac{\partial G}{\partial t}(z,t) = r_{burst}\left(p\,\frac{1}{1-(1-p)z} - 1\right)G(z,t) + r_{deg}(1-z)\frac{\partial G}{\partial z}(z,t).$$

Taken together, the result is a PDE of order one and equivalent to

$$\begin{aligned}G(z,t) =&\frac{1-(1-p)z}{r_{burst}p - r_{burst}(1-(1-p)z)}\frac{\partial G}{\partial t}(z,t)\\ &- \frac{r_{deg}(1-z)(1-(1-p)z)}{r_{burst}p - r_{burst}(1-(1-p)z)}\frac{\partial G}{\partial z}(z,t).\end{aligned} \tag{4.13}$$

In the following we show how to solve the PDE

$$G(z(y),t(y)) = G_z\dot{z} + G_t\dot{t}.$$

Ansatz: $G(z,t) = U(x,w) = U_z\dot{z} + U_t\dot{t}$

To use this ansatz, we need to determine x and w. We read $\dot{z}$ and $\dot{t}$ from the full equation given by (4.13):

$$\begin{aligned}
\dot{z} &= -\frac{r_{deg}(1-z)(1-(1-p)z)}{r_{burst}p - r_{burst}(1-(1-p)z)}\\
\dot{t} &= \frac{1-(1-p)z}{r_{burst}p - r_{burst}(1-(1-p)z)}\\
\frac{\dot{z}}{\dot{t}} &= \frac{\frac{dz}{dy}}{\frac{dt}{dy}} = \frac{dz}{dt} = \frac{-r_{deg}(1-z)(1-(1-p)z)(r_{burst}p - r_{burst}(1-(1-p)z)}{(r_{burst}p - r_{burst}(1-(1-p)z))(1-(1-p)z)} = -r_{deg}(1-z).
\end{aligned}$$

Thus, it follows that

$$dt = \frac{dz}{r_{deg}(z-1)}.$$

Integrating both sides yields

$$\int dt = \int \frac{1}{r_{deg}(z-1)}dz \qquad \Leftrightarrow \qquad \log(z-1) = r_{deg}t + \tilde{c}$$

for an arbitrary constant $\tilde{c}$. Next, we take the exponential of both sides

$$z-1 = ce^{r_{deg}t} \qquad \Leftrightarrow \qquad c = (z-1)e^{-r_{deg}t}$$

for a constant c. Choose $x = c = (z-1)e^{-r_{deg}t}$ and $w = t$. Then it follows that $z = xe^{r_{deg}w} + 1$. For the derivatives, we obtain

$$\begin{aligned}
x_z &= e^{-r_{deg}t} & x_t &= -(z-1)r_{deg}e^{-r_{deg}t}\\
w_z &= 0 & w_t &= 1.
\end{aligned}$$

Next, we need to determine U_z and U_t:

$$\begin{aligned}
U_z &= U_x x_z + U_w w_z = e^{-r_{deg}t}U_x\\
U_t &= U_x x_t + U_w w_t = -r_{deg}e^{-r_{deg}t}(z-1)U_x + U_w.
\end{aligned}$$

Finally, we can compute U:

$$\begin{aligned}
U(x,w) &= U_z\dot{z} + U_t\dot{t}\\
&= e^{-r_{deg}t}U_x\frac{(-r_{deg})(1-z)(1-(1-p)z)}{r_{burst}p - r_{burst}(1-(1-p)z)}\\
&\qquad + \frac{1-(1-p)z}{r_{burst}p - r_{burst}(1-(1-p)z)}\left(r_{deg}e^{-r_{deg}t}(1-z)U_x + U_w\right)\\
&= \frac{1-(1-p)z}{r_{burst}p - r_{burst}(1-(1-p)z)}U_w.
\end{aligned}$$

Plug in z and t to get U only in terms of x and w:

$$U(x,w) = \frac{1-(1-p)(xe^{r_{deg}w}+1)}{r_{burst}p - r_{burst}(1-(1-p)(xe^{r_{deg}w}+1))}U_w$$

$$= \frac{1 - xe^{r_{deg}w} - 1 + pxe^{r_{deg}w} + p}{r_{burst}p - r_{burst} + r_{burst}xe^{r_{deg}w} + r_{burst} - r_{burst}pxe^{r_{deg}w} - r_{burst}p} U_w$$
$$= \frac{e^{r_{deg}w}(px - x + pe^{-r_{deg}w})}{e^{r_{deg}w}(r_{burst}x - r_{burst}px)} U_w.$$

As $U_w = dU/dw$, we can separate the terms depending on U and the terms depending on w:

$$\frac{dw}{px - x + pe^{-r_{deg}w}} = \frac{dU}{U(r_{burst}x - r_{burst}px)}.$$

Integrating both sides leads to:

$$-\frac{\log\left(pxe^{r_{deg}w} - xe^{r_{deg}w} + p\right)}{r_{deg}x - r_{deg}px} = \frac{\log(U)}{r_{burst}x - pr_{burst}x} + f(x),$$

where $f(x)$ is seen as a constant with respect to w and U and thus can only be a function that depends on x. Then

$$\log(U) = -\frac{r_{burst}x(1-p)}{r_{deg}x(1-p)} \log\left(pxe^{r_{deg}w} - xe^{r_{deg}w} + p\right) + f(x).$$

Next, we take the exponential on both sides

$$U = \left(pxe^{r_{deg}w} - xe^{r_{deg}w} + p\right)^{-\frac{r_{burst}}{r_{deg}}} f(x) = \left(-xe^{r_{deg}w}(1-p) + p\right)^{-\frac{r_{burst}}{r_{deg}}} f(x)$$

Return to the parameterization in terms of z and t:

$$G(z,t) = U(x = (z-1)e^{-r_{deg}t}, w = t)$$
$$= \left(-(z-1)e^{-r_{deg}t}e^{r_{deg}t}(1-p) + p\right)^{-\frac{r_{burst}}{r_{deg}}} f((z-1)e^{-r_{deg}t}),$$

where $f((z-1)e^{-r_{deg}t}) =: f(z,t)$ now represents a function that depends on z and t. We get

$$G(z,t) = (-z + zp + 1 - p + p)^{-\frac{r_{burst}}{r_{deg}}} f(z,t)$$
$$= (1 - z(1-p))^{-\frac{r_{burst}}{r_{deg}}} f(z,t).$$

The right hand side is of the form of the probability generating function of a NB distribution with parameters r_{NB} and p_{NB} as stated in Definition A.8 if one chooses $f(z,t) = p_{\text{NB}}^{r_{\text{NB}}}$, $r_{\text{NB}} = r_{burst}/r_{deg}$ and $p_{\text{NB}} = p$. Since the mean burst size in the bursting model is s_{burst}, the parameter p of the geometric distribution and hence the parameter p_{NB} of the NB distribution is equal to $(1 + s_{burst})^{-1}$.

4.4 Further Distributions and Models

In the previous section, we showed how some of the most known gene transcription models fit into the generalized framework and how a steady state distribution can be connected via OU processes to the intensity process of the compound Poisson distribution. This intensity process gives already hints on how the underlying biological transcription model looks like. In this section we investigate further distributions and models.

4.4.1 Basic-Bursting Model

Another obvious consideration is to check what happens when combining two of the previously discussed models. Here, we calculate the resulting steady state distribution when combining the basic and the bursting model, depicted in Figure 4.8. We call the resulting model basic-bursting model.

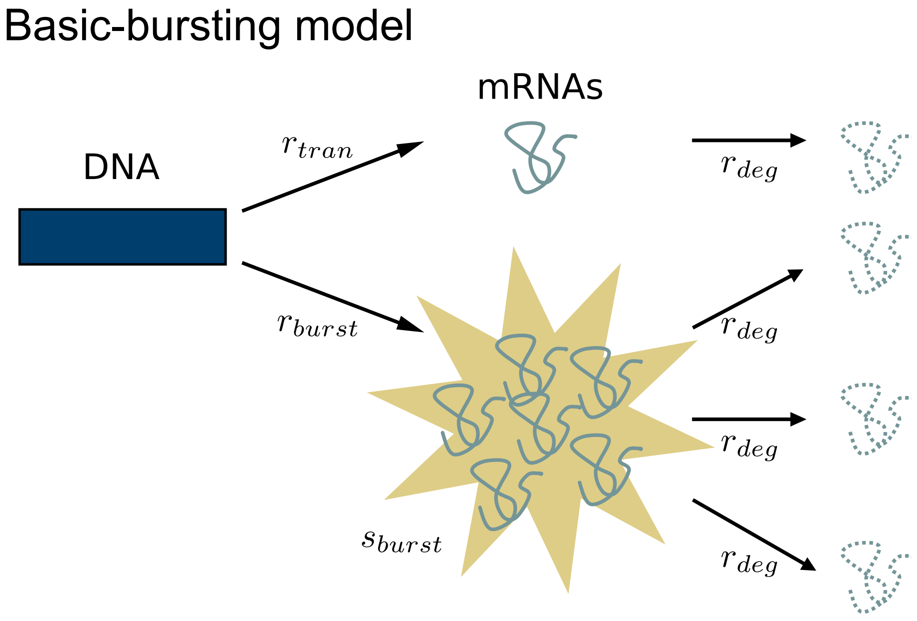


Figure 4.8: Basic-bursting model of gene expression consists of constant and bursty transcription and constant degradation of mRNA

We do this along the line of the calculations in Appendix D.1 and Section 4.3.1.3.

$$\begin{aligned}\frac{d\mathcal{P}(n,t)}{dt} &= \sum_{x=0}^{\infty} r_{burst}\mathcal{P}(n-x,t)\mathrm{P}_{Geo}(X=x) + r_{tran}\mathcal{P}(n-1,t) + r_{deg}(n+1)\mathcal{P}(n+1,t)\\ &\quad - \left(\sum_{x=0}^{\infty} r_{burst}\mathrm{P}_{Geo}(X=x) + r_{tran} + r_{deg}\, n\right)\mathcal{P}(n,t),\end{aligned}$$

where P denotes the probability mass function of a random variable X that is geometrically distributed with success probability p. The derivative of the probability-generating function then reads

$$\begin{aligned}\frac{\partial G}{\partial t}(z,t) &= \sum_{n=0}^{\infty} z^n \frac{d\mathcal{P}(n,t)}{dt}\\ &= \sum_{x=0}^{\infty} z^x r_{burst}\mathrm{P}_{Geo}(X=x)\sum_{n=0}^{\infty} z^{n-x}\mathcal{P}(n-x,t) + \sum_{n=0}^{\infty} z^n r_{tran}\mathcal{P}(n-1,t)\\ &\quad + \sum_{n=0}^{\infty} r_{deg}(n+1)z^n\mathcal{P}(n+1,t) - \sum_{x=0}^{\infty} r_{burst}\mathrm{P}_{Geo}(X=x)\sum_{n=0}^{\infty} z^n\mathcal{P}(n,t)\end{aligned}$$

$$
\begin{aligned}
&- \sum_{n=0}^{\infty} r_{tran} z^n \mathcal{P}(n,t) - \sum_{n=0}^{\infty} r_{deg} n z^n \mathcal{P}(n,t) \\
=& \, r_{burst} \sum_{x=0}^{\infty} z^x \mathrm{P}_{Geo}(X=x) \sum_{n=0}^{\infty} z^{n-x} \mathcal{P}(n-x,t) + r_{tran} z \sum_{n=0}^{\infty} z^{n-1} \mathcal{P}(n-1,t) \\
&+ r_{deg} \sum_{n=0}^{\infty} (n+1) z^n \mathcal{P}(n+1,t) - r_{burst} \sum_{x=0}^{\infty} \mathrm{P}_{Geo}(X=x) \sum_{n=0}^{\infty} z^n \mathcal{P}(n,t)
\end{aligned}
$$

With

$$
\begin{aligned}
G(z,t) &= \sum_{n=0}^{\infty} z^{n-x} \mathcal{P}(n-x,t) \\
G(z,t) &= \sum_{n=0}^{\infty} z^{n} \mathcal{P}(n,t) \\
\frac{\partial G}{\partial z}(z,t) &= \sum_{n=0}^{\infty} n z^{n-1} \mathcal{P}(n,t) \\
\frac{\partial G}{\partial z}(z,t) &= \sum_{n=0}^{\infty} (n+1) z^{n} \mathcal{P}(n+1,t)
\end{aligned}
$$

it follows that

$$
\begin{aligned}
\frac{\partial G}{\partial t}(z,t) = &\left[r_{burst} \left(\sum_{x=0}^{\infty} z^x \mathrm{P}_{Geo}(X=x) - \sum_{x=0}^{\infty} \mathrm{P}_{Geo}(X=x) \right) + r_{tran}(z-1) \right] G(z,t) \\
&+ r_{deg}(1-z) \frac{\partial G}{\partial z}(z,t).
\end{aligned}
$$

Because of $\mathrm{P}_{Geo}(X=x) = (1-p)^x p$ and $\sum_{x=0}^{\infty} \mathrm{P}_{Geo}(X=x) = 1$, it follows that

$$
\begin{aligned}
\frac{\partial G}{\partial t}(z,t) = &\left[r_{burst} \left(p \, \frac{1}{1-(1-p)z} - 1 \right) + r_{tran}(z-1) \right] G(z,t) \\
&+ r_{deg}(1-z) \frac{\partial G}{\partial z}(z,t).
\end{aligned}
$$

Taken together, the result is a PDE of order one and equivalent to

$$
\begin{aligned}
G(z,t) = &\frac{1-z(1-p)}{r_{burst} p - r_{burst}(1-z(1-p)) + r_{tran}(z-1)(1-z(1-p))} \frac{\partial G}{\partial t}(z,t) \\
&- \frac{r_{deg}(1-z)(1-z(1-p))}{r_{burst} p - r_{burst}(1-(1-p)z) + r_{tran}(z-1)(1-z(1-p))} \frac{\partial G}{\partial z}(z,t).
\end{aligned}
\tag{4.14}
$$

In the following we show how to solve the PDE

$$
G(z(y), t(y)) = G_z \dot{z} + G_t \dot{t}.
$$

Ansatz: $G(z,t) = U(x,w) = U_z\dot{z} + U_t\dot{t}$

To use this ansatz, we need to determine x and w. We read $\dot{z}$ and $\dot{t}$ from the full equation given by (4.14):

$$\begin{aligned}
\dot{z} &= -\frac{r_{deg}(1-z)(1-z(1-p))}{r_{burst}p - r_{burst}(1-z(1-p)) + r_{tran}(z-1)(1-z(1-p))}\\
\dot{t} &= \frac{1-z(1-p)}{r_{burst}p - r_{burst}(1-z(1-p)) + r_{tran}(z-1)(1-z(1-p))}\\
\frac{\dot{z}}{\dot{t}} &= \frac{\frac{dz}{dy}}{\frac{dt}{dy}} = \frac{dz}{dt} = \frac{-r_{deg}(1-z)(1-z(1-p))}{(1-z(1-p))} = -r_{deg}(1-z).
\end{aligned}$$

Thus, it follows that

$$dt = \frac{dz}{r_{deg}(z-1)}.$$

Integrating both sides yields

$$\int dt = \int \frac{1}{r_{deg}(z-1)}dz \qquad \Leftrightarrow \qquad \log(z-1) = \; r_{deg}t + \tilde{c}.$$

for an arbitrary constant $\tilde{c}$. Next, we take the exponential of both sides

$$z - 1 = \; ce^{r_{deg}t} \qquad \Leftrightarrow \qquad c = \; (z-1)e^{-r_{deg}t}$$

for a constant c. Choose $x = \; c = (z-1)e^{-r_{deg}t}$ and $w = \; t$. Then it follows that $z = xe^{r_{deg}w} + 1$. For the derivatives, we obtain

$$\begin{aligned}
x_z &= e^{-r_{deg}t} & x_t &= -(z-1)r_{deg}e^{-r_{deg}t}\\
w_z &= 0 & w_t &= 1.
\end{aligned}$$

Next, we need to determine U_z and U_t:

$$\begin{aligned}
U_z &= U_x x_z + U_w w_z = e^{-r_{deg}t}U_x\\
U_t &= U_x x_t + U_w w_t = -r_{deg}e^{-r_{deg}t}(z-1)U_x + U_w.
\end{aligned}$$

Finally, we can compute U:

$$\begin{aligned}
U(x,w) =& U_z\dot{z} + U_t\dot{t}\\
=& e^{-r_{deg}t}U_x\dot{z} - (z-1)r_{deg}e^{-r_{deg}t}U_x\frac{\dot{z}}{-r_{deg}(1-z)} + \frac{\dot{z}}{-r_{deg}(1-z)}U_w\\
=& \frac{\dot{z}}{-r_{deg}(1-z)}U_w \; = \dot{}tU_w\\
=& \frac{1-z(1-p)}{r_{burst}p - r_{burst}(1-z(1-p)) + r_{tran}(z-1)(1-z(1-p))}U_w
\end{aligned}$$

Fill in z and t in terms of x and w.

$$\begin{aligned}
&=\frac{(1-(xe^{r_{deg}w}+1)(1-p))U_w}{r_{burst}p-r_{burst}(1-(xe^{r_{deg}w}+1)(1-p))+r_{tran}(xe^{r_{deg}w}+1-1)(1-(xe^{r_{deg}w}+1)(1-p))}\\
&=\frac{(-xe^{r_{deg}w}+pxe^{r_{deg}w}+p)U_w}{r_{burst}p-r_{burst}(p+pxe^{r_{deg}w}-xe^{r_{deg}w})+r_{tran}xe^{r_{deg}w}(p+pxe^{r_{deg}w}-xe^{r_{deg}w})}\\
&=\frac{U_w}{r_{burst}p(p+pxe^{r_{deg}w}-xe^{r_{deg}w})^{-1}-r_{burst}+r_{tran}xe^{r_{deg}w}}
\end{aligned}$$

As $U_w = dU/dw$, terms depending on U and w can be separated:

$$\frac{dU}{U}=\left(\frac{r_{burst}p}{p+pxe^{r_{deg}w}-xe^{r_{deg}w}}-r_{burst}+r_{tran}xe^{r_{deg}w}\right)dw.$$

Integrating both sides leads to:

$$\log(U)=r_{burst}p\left(\frac{r_{deg}w-\log\left(p+pxe^{r_{deg}w}-xe^{r_{deg}w}\right)}{pr_{deg}}\right)-r_{burst}w+\frac{r_{tran}xe^{r_{deg}w}}{r_{deg}}+f(x),$$

where $f(x)$ is seen as a constant with respect to w and U and thus can only be a function that depends on x. Then

$$=r_{burst}w-\frac{r_{burst}}{r_{deg}}\log\left(p+pxe^{r_{deg}w}-xe^{r_{deg}w}\right)-r_{burst}w+\frac{r_{tran}xe^{r_{deg}w}}{r_{deg}}+f(x).$$

Next, we take the exponential on both sides

$$U=\left(p+pxe^{r_{deg}w}-xe^{r_{deg}w}\right)^{-\frac{r_{burst}}{r_{deg}}}\exp\left(\frac{r_{tran}xe^{r_{deg}w}}{r_{deg}}\right)f(x).$$

Return to the parameterization in terms of z and t:

$$\begin{aligned}
G(z,t)&=U(x=(z-1)e^{-r_{deg}t},w=t)\\
&=\left(p+p(z-1)-(z-1)\right)^{-\frac{r_{burst}}{r_{deg}}}\exp\left(\frac{r_{tran}(z-1)}{r_{deg}}\right)f((z-1)e^{-r_{deg}t}),
\end{aligned}$$

where $f((z-1)e^{-r_{deg}t}) =: f(z,t)$ now represents a function that depends on z and t. We get

$$\begin{aligned}
G(z,t)&=\left(p+pz-p-z+1\right)^{-\frac{r_{burst}}{r_{deg}}}\exp\left(\frac{r_{tran}(z-1)}{r_{deg}}\right)f(z,t)\\
&=\left(1-z(1-p)\right)^{-\frac{r_{burst}}{r_{deg}}}\exp\left(\frac{r_{tran}(z-1)}{r_{deg}}\right)f(z,t).
\end{aligned}$$

With Definition A.7, we know that

$$G_{\mathrm{Pois}\left(\frac{r_{tran}}{r_{deg}}\right)}(z)=\exp\left(\frac{r_{tran}(z-1)}{r_{deg}}\right)$$

is the probability generating function of a Poisson distribution with parameter $\frac{r_{tran}}{r_{deg}}$. Analogously to the master equation of the bursting model in Section 4.3.1.3, we choose $f(z,t) = p_{\mathrm{NB}}^{r_{\mathrm{NB}}}$, $r_{\mathrm{NB}} = r_{burst}/r_{deg}$ and $p_{\mathrm{NB}} = p = (1+s_{burst})^{-1}$. Then here again, the remaining parts describe the probability generating function of a NB distribution

$$G_{\mathrm{NB}\left(p,\frac{r_{burst}}{r_{deg}}\right)}(z) = \left(\frac{(1+s_{burst})^{-1}}{(1-z(1-(1+s_{burst})^{-1})}\right)^{\frac{r_{burst}}{r_{deg}}}.$$

Taken together $G(z,t) = G(z)$ is time-independent and the product of the two probability generating functions $G_{\mathrm{Pois}\left(\frac{r_{tran}}{r_{deg}}\right)}(z)$ and $G_{\mathrm{NB}\left(\frac{r_{burst}}{r_{deg}},(1+s_{burst})^{-1}\right)}(z)$. This results in $G(z)$ describing the probability generating function of some random variable Z that is the sum of two independent random variables X and Y, where $X \sim \mathrm{Pois}\left(\frac{r_{tran}}{r_{deg}}\right)$ and $Y \sim \mathrm{NB}\left(\frac{r_{burst}}{r_{deg}}, (1+s_{burst})^{-1}\right)$. This means that the steady state distribution of the basic-bursting model is exactly the sum of the steady state distributions of the individual processes.

We know from Example 3.12 that the convolution of a Poisson and a NB distribution results in a Delaporte (DEL) distribution, given in Definition A.10. With this $G(z)$ describes the probability generating function of

$$Z \sim \mathrm{DEL}\left(\frac{r_{tran} + r_{burst}s_{burst}}{r_{deg}}, \frac{r_{deg}}{r_{burst}}, \frac{r_{tran}}{r_{tran} + r_{burst}s_{burst}}\right).$$

Taken together, the steady state distribution of mRNA counts that are created in a basic-bursting transcription process is a DEL distribution. Constructions like this might be interesting when modeling for example biallelic gene expression, i. e. both alleles transcribe independent but with different underlying mechanistic processes.

4.4.2 Poisson-Inverse Gaussian Distribution

Another interesting distribution to model count data is the Poisson-inverse Gaussian (PIG) distribution (see Example 3.5). This distribution is used to model for example species abundances (Ord and Whitmore, 1986) or in assurance modeling (Zha et al., 2016).

The aim of this section is to see if we can derive a possible underlying biological transcription model if we assume mRNA to follow in steady state the PIG distribution. This is done in two steps, analogously to the derivation of the underlying transcription model of the NB distribution in Section 4.3.1. First, using the compounding distribution, the corresponding subordinator of the intensity process I_t given by the OU process in Equation (4.10) will be derived. Then the subordinator is used to come up with possible mechanisms that might describe the underlying transcription process.

We start with the compounding distribution of the target distribution which is here the PIG distribution. We want to derive the subordinator on the OU process (4.10) under the assumption that in steady state the intensity I is modeled by an inverse Gaussian

distribution (see Definition A.2). The IG distribution fulfills the prerequisites to apply Lemma 4.1 and therefore the three step approach (described in Section 4.3) can be applied with $\mathcal{D}$ being the IG distribution. With this the corresponding subordinator can be easily calculated.
The characteristic function of $I \sim IG_{\lambda,\mu}$ is given by (see Definition A.2)

$$\begin{aligned}\varphi_I(z) &= \exp\left(\frac{\lambda}{\mu}\left(1-\sqrt{1-\frac{2\mu^2 iz}{\lambda}}\right)\right)\\ &= \exp\left(\frac{\lambda}{\mu}\int_0^z\left(-\frac{1}{2}\left(1-\frac{2\mu^2 ix}{\lambda}\right)^{-\frac{1}{2}}\left(\frac{2\mu^2 i}{\lambda}\right)\right)dx\right)\\ &= \exp\left(r_{deg}^{-1}\int_0^z x\ \mu i r_{deg}\left(1-\frac{2\mu^2 ix}{\lambda}\right)^{-\frac{1}{2}}\frac{1}{x}dx\right)\\ &= \exp\left(r_{deg}^{-1}\int_0^z \phi(x)x^{-1}dx\right),\end{aligned}$$

with $\phi(x) = x\mu i r_{deg}\left(1-\frac{2\mu^2 ix}{\lambda}\right)^{-\frac{1}{2}}$.

The next step is to find the corresponding subordinator L_t. For this, the characteristic function of L_1 is given by:

$$\begin{aligned}\varphi_{L_1}(z) &= \exp(\phi(z)) = \exp\left(z\mu i r_{deg}\left(1-\frac{2\mu^2 iz}{\lambda}\right)^{-\frac{1}{2}}\right)\\ &= \exp\left(\frac{r_{deg}\mu iz}{\sqrt{1-\frac{2\mu^2 iz}{\lambda}}}\right) = \exp\left(\frac{r_{deg}\lambda}{2\mu}\left(\frac{-1+\frac{2iz\mu^2}{\lambda}+1}{\sqrt{1-\frac{2\mu^2 iz}{\lambda}}}\right)\right)\\ &= \exp\left(\frac{r_{deg}\lambda}{2\mu}\left(1-\sqrt{1-\frac{2\mu^2 iz}{\lambda}}-1+\frac{1}{\sqrt{1-\frac{2\mu^2 iz}{\lambda}}}\right)\right)\\ &= \exp\left(\frac{r_{deg}\lambda}{2\mu}\left(1-\sqrt{1-\frac{2\mu^2 iz}{\lambda}}\right)\right)\exp\left(\frac{r_{deg}\lambda}{2\mu}\left(\left(1-\frac{2\mu^2 iz}{\lambda}\right)^{-\frac{1}{2}}-1\right)\right)\\ &= \varphi_{Y_1}(z)\varphi_{J_1}(z)\end{aligned}$$

This shows that the characteristic function of L_1 is a product of two characteristic functions. Therefore the subordinator $L_t = Y_t + J_t$ is the sum of two subordinators Y_t and J_t. In fact, $\varphi_{Y_1}(z)$ and $\varphi_{J_1}(z)$ are known characteristic functions:

- $\varphi_{Y_1}(z)$ is the characteristic function at $t = 1$ of an IG process (Definition 3.12) with mean parameter $\mu_{IGP} = \frac{r_{deg}\mu}{2}$ and shape parameter $\lambda_{IGP} = \frac{r_{deg}^2\lambda}{4}$.
- $\varphi_{J_1}(z)$ is the characteristic function at $t = 1$ of a CPP (Definition 3.11) with $\frac{\lambda r_{deg}}{2\mu}$ the waiting time parameter and $\Gamma\left(\frac{1}{2}, \frac{\lambda}{2\mu^2}\right)$ distributed jump sizes.

This connection between the IG distribution and the subordinator, that consists of the sum of a IG process and a CPP with gamma distributed jump sizes has been inferred before using Lévy densities (Barndorff-Nielsen and Shephard, 2001a, Valdivieso et al., 2009). In our calculation we used the three step method described in Section 4.3 using characteristic functions.

We have found the subordinator and thus the corresponding SDE. This does not mean that the corresponding mechanistic process has been identified. In contrary to the new first part of the subordinator – the IG process –, the second part of the subordinator – the new CPP subordinator – is not entirely unknown to us as it is similar to the CPP subordinator that we already know from the bursting model in Section 4.3.1.1. To infer a plausible biological transcription process which we will call IG bursting model from the new subordinator, we need to understand what this sum of subordinators means. From the basic-bursting model in Section 4.4.1 we know that the transcription process can consist of two independent parts whereas degradation is not changed. In the following we will look at both parts of the subordinator separately.

Note that all parameters above are described in terms of the parameters of the target IG distribution λ and μ and the degradation rate r_{deg}. This means we have three parameters in the model – two in addition to the degradation rate r_{deg}. When inferring a new transcription model, we want to use the parameters of this mechanistic model and describe the IG steady state distribution using these. Since the subordinator splits in two parts and our goal is to infer two (independent) transcription processes, we will use one model parameter for each part. So we will use μ_R for the first part, as the mean transcription rate and r_{burst} as the rate parameter for a possible second bursty part.

4.4.2.1 IG Subordinator

We start with the unknown IG subordinator (see Definition 3.12) and compare it to the subordinator of the basic model. The IG subordinator has increments (= jumps) $Y_t - Y_s$ which are IG $(\mu_R(t-s), \lambda_R(t-s)^2)$ distributed for all $t > s > 0$. Hence for $r_{tran} = \mu_R$ the mean function of the IG process is the same as the basic subordinator. Comparing this with the constant transcription rate r_{trans} of the basic process, we propose as first part of the new IG bursting model a modified basic model, where the constant transcription rate is replaced by a random variable R that follows an IG $(\mu_R, \mu_R r_{burst})$ distribution. This means that the transcription rate is not fixed in a cell over time, but at each event a different transcription rate has to be taken into account. This is depicted in Figure 4.9.

4.4.2.2 New CPP Subordinator

In Section 4.3.1.1, the CPP subordinator L_t^{burst} has jumps that are exponentially distributed and lead to a bursty transcription process with geometrically distributed burst sizes. The new CPP has jumps that follow the more general gamma distribution. However, this can be analogously translated into a mechanistic bursting process with

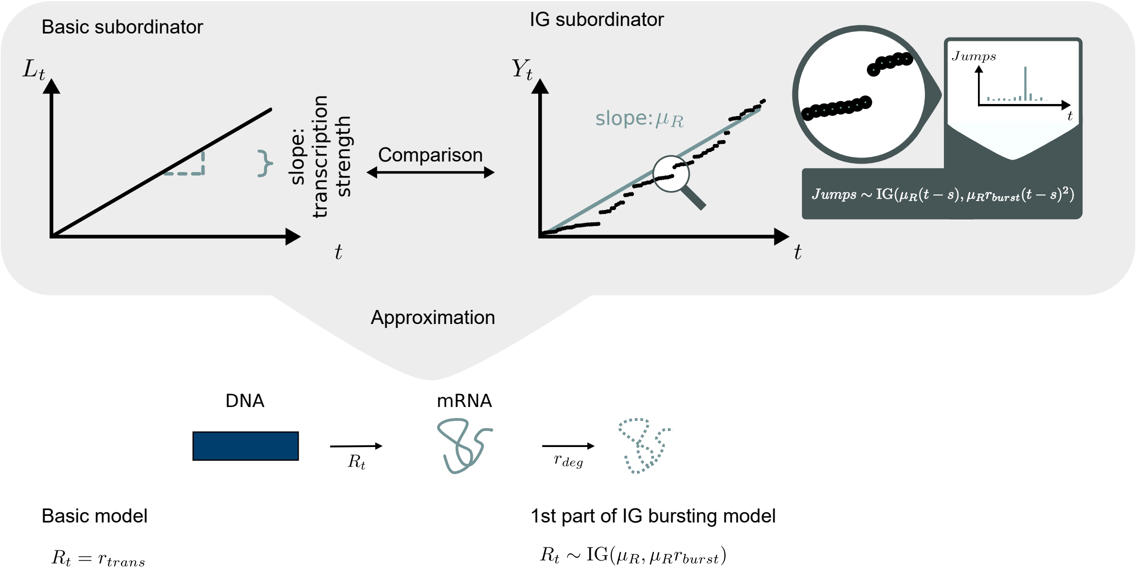

Basic subordinator
L_t
slope: transcription strength
t
Comparison
IG subordinator
Y_t
slope: μ_R
t
$Jumps$
t
$Jumps \sim \mathrm{IG}(\mu_R(t-s), \mu_R r_{burst}(t-s)^2)$
Approximation
DNA
mRNA
R_t
r_{deg}
Basic model
$R_t = r_{trans}$
1st part of IG bursting model
$R_t \sim \mathrm{IG}(\mu_R, \mu_R r_{burst})$

Figure 4.9: Comparison of the IG subordinator with the subordinator of the basic model. The mean function of the IG subordinator $\mu_R t$ is exactly the basic subordinator if $r_{tran} = \mu_R$. The IG subordinator is a pure jump process, where the jumps are $\mathrm{IG}(\mu_R(t-s), \mu_R r_{burst}(t-s)^2)$ distributed. Comparing these increments with the constant rate of the basic process, we propose as second part of the new IG bursting model, a modified basic model where the transcription rate R_t of mRNAs is not a constant but follows an $\mathrm{IG}(\mu_R, \mu_R r_{burst})$ distribution. Note that this rate is not fixed for a cell over time, but continuously newly drawn transcription rates have to be taken into account.

NB distributed burst sizes. The bursts occur with $\mathrm{EXP}(r_{burst})$ distributed times but the bursts S follow a NB $\left(\frac{1}{2}, r_{burst}/\left(r_{burst} + 2\mu_R\right)\right)$ distribution.
Comparing these we see that both CPP are similar, as both have Γ distributed jump sizes. In fact, L_t^{burst} had exponentially distributed jump sizes which are the same as $\Gamma(\alpha = 1, \lambda)$ (see Definition A.3). Therefore we can also generalize the resulting biological burst process. In the bursting model in Section 4.3.1.2 the burst sizes follow a geometric distribution. The geometric distribution is a special case of the NB distribution. In detail $\mathrm{Geo}(p) = \mathrm{NB}(r = 1, p)$. So with respect to the parameters in the new CPP subordinator, we generalize the burst process to a new burst process. In this new burst process, waiting times are still $\mathrm{EXP}(r_{burst})$ distributed. Bursts follow now a NB distribution, whose parameters are determined by the gamma distribution of the Γ distributed jumps sizes of the new CPP subordinator. Therefore the parameters of the NB bursts are $r = \frac{1}{2}$ and $p = \frac{r_{burst}}{r_{burst}+2\mu_R}$. Like this the mean burst size $\frac{\mu_R}{r_{burst}}$ is the same for the new CPP subordinator and the new bursting model. This derivation is also depicted in Figure 4.10.

4.4.2.3 The IG Bursting Model

With this reasoning, combining the two proposed parts to one model, an actual underlying process resulting in a PIG steady state distribution could look like the one shown in Figure 4.11. We are calling this model IG bursting model.

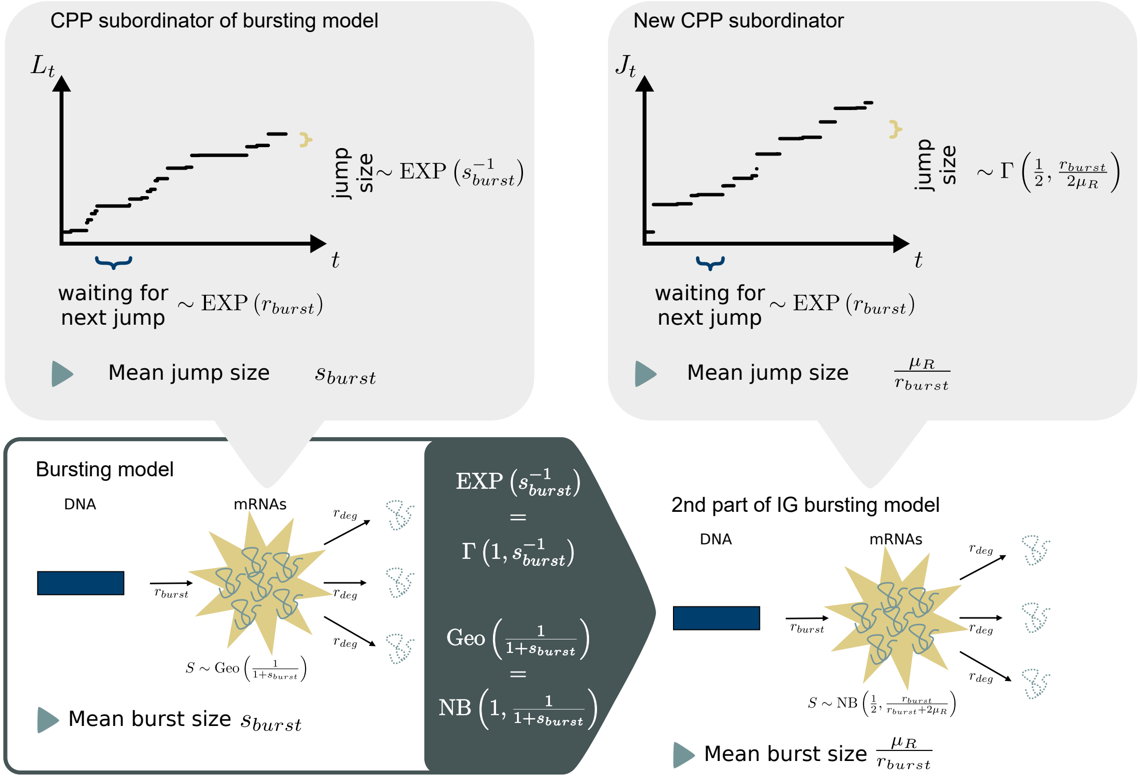
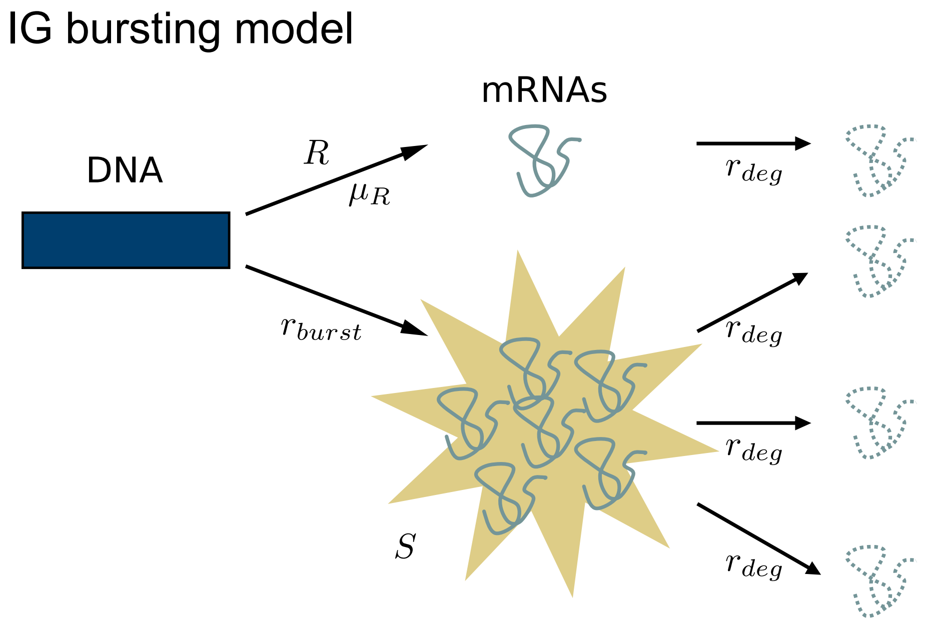

CPP subordinator of bursting model
L_t
t
jump size
$\sim \text{EXP}\left(s_{burst}^{-1}\right)$
waiting for next jump
$\sim \text{EXP}\left(r_{burst}\right)$
Mean jump size s_{burst}
New CPP subordinator
J_t
t
jump size
$\sim \Gamma\left(\frac{1}{2}, \frac{r_{burst}}{2\mu_R}\right)$
waiting for next jump
$\sim \text{EXP}\left(r_{burst}\right)$
Mean jump size $\frac{\mu_R}{r_{burst}}$
Bursting model
DNA
mRNAs
r_{burst}
r_{deg}
r_{deg}
r_{deg}
$S \sim \text{Geo}\left(\frac{1}{1+s_{burst}}\right)$
Mean burst size s_{burst}
$\text{EXP}\left(s_{burst}^{-1}\right) = \Gamma\left(1, s_{burst}^{-1}\right)$
$\text{Geo}\left(\frac{1}{1+s_{burst}}\right) = \text{NB}\left(1, \frac{1}{1+s_{burst}}\right)$
2nd part of IG bursting model
DNA
mRNAs
r_{burst}
r_{deg}
r_{deg}
r_{deg}
$S \sim \text{NB}\left(\frac{1}{2}, \frac{r_{burst}}{r_{burst}+2\mu_R}\right)$
Mean burst size $\frac{\mu_R}{r_{burst}}$

Figure 4.10: Comparison of the new CPP subordinator with the CPP subordinator of the bursting model. The CPP differ in their jump distributions. In the CPP of the bursting model this intensity was exponentially distributed with mean size s_{burst}. In the new CPP model jump sizes follow a gamma distribution, where the first parameter is 1 and the second parameter is given by $r_{burst}/(2\mu_R)$. From the Section 4.3.1.2 we know that this CPP as subordinator in Equation (4.10) leads to the depicted bursting process. The exponential distribution is a special form of the gamma distribution as well as the geometric distribution is a special form of the NB distribution. Using this, we propose for the 2nd part a possible new transcription model – called IG bursting model – to consist of a burst with the depicted rates.

Note that both parts are not independent of the other part, as they use both the μ_R and r_{burst} parameters. Only together they result in a model that is a possible underlying model, where the steady state distribution then follows a PIG $\left(\frac{2\mu_R}{r_{deg}}, \frac{4r_{burst}\mu_R}{r_{deg}^2}\right)$. Note that, analogously to the previous models, the three IG bursting model parameters are not identifiable. We can only infer $\frac{\mu_R}{r_{deg}}$ and $\frac{r_{burst}}{r_{deg}}$.

4.5 Application: Data Generation and Model Selection

In this section we show how the distribution models can be applied to real-world data. To do so, we need to consider the real-world conditions and include them in the distribution models. First, we specify the adjustments that need to be made.



$$R \sim \mathrm{IG}\left(\mu_R, \mu_R r_{burst}\right)$$
$$S \sim \mathrm{NB}\left(\frac{1}{2}, \frac{r_{burst}}{r_{burst}+2\mu_R}\right)$$

Figure 4.11: IG bursting model of gene expression consists of an IG distributed transcription rate that generates one mRNA at a time and of a parallel bursty transcription, both transcription events are determined by the mean parameter of the IG distribution and the rate parameter of the bursts. The degradation rate of mRNA remains constant.

Then we describe our implementation in the R package **scModels**, which we use for the distribution fitting. In addition, we have also implemented the five transcription models proposed above so that we can generate data via their respective Gillespie algorithm (Gillespie, 1976). Using these we present a simulation study in which we generate perfect-world data with all five transcription models (i. e. via the respective Gillespie algorithm without any technical or biological noise) and fit the presented distributions to the data. By model selection we want to investigate which distribution assumption fits most often to the respective transcription models and which distribution could be the best general choice. Finally, we will model real-world data by the modified distributions. We want to asses if the selected distributions also perform best here, and we will investigate if we can find some gene traits that allow simpler (or more complicated) distribution models.

4.5.1 Heterogeneity and Dropout

The transcription and degradation models considered so far describe the number of mRNA molecules for homogeneously expressed genes that are actually present in a cell. Real-world data is usually more complex: First, cell populations may be heterogeneous in their gene expression. Second, scRNA-seq measurements will be subject to measurement error. For example, they often contain a large number of zeros. Dropouts are zero values caused by technical errors. Thus, in reality some mRNA was present, but due to some circumstances the measurement is zero. Regardless of what causes this phenomenon, a data model should take this property into account. Therefore we describe three model extensions, one for zero inflation, one for mixing two distributions and the third for zero inflation and simultaneous mixing of two distributions.

Although $\mathcal{D}$ can in general be every possible distribution, we will discuss below the Pois, NB, PB, DEL and PIG distribution, as we have done in the previous sections. Data that originates from different cell populations (in terms of different transcriptomic properties) can be modeled mathematically by mixture distributions, as introduced in Section 3.1.1. We will use the case described in Example 3.1, where a population consists of two subpopulations and each of them is modeled by one single distribution. Here, we assume that both populations follow the same distribution $\mathcal{D}$ with PMF f but are parameterized with different parameters θ_1 and θ_2. Therefore, the corresponding mixture density is given by

$$f_{\text{2mix}}(x|\theta_1, \theta_2, p) = p\, f(x|\theta_1) + (1-p) f(x|\theta_2),$$

where p describes the share between the two distributions.

An appropriate model for the occurrence of the above-mentioned dropout is a zero-inflated distribution (see Equation (3.1) and Kharchenko et al., 2014). We add zero inflation to a homogeneous population, which is described by the discrete count distribution $\mathcal{D}$ with PMF f and parameter θ, as shown in Example 3.2. With this the resulting PMF is given by

$$f_{\text{zi}}(x|\theta, p) = p\, \mathbb{1}_{\{0\}}(x) + (1-p) f(x|\theta),$$

where $\mathbb{1}_{\{0\}}(x)$ is the indicator function with point mass at zero and f is the PMF of the selected distribution $\mathcal{D}$.

Analogously these two mixtures can be combined so that zero-inflation is added to a mixture of several distributions. This leads to mRNA counts being modeled by

$$f_{\text{zi-2mix}}(x|\theta_1, \theta_2, p_1, p_2) = p_1 \mathbb{1}_{\{0\}}(x) + p_2\, f(x|\theta_1) + (1 - p_1 - p_2)\, f(x|\theta_2),$$

where p_1, p_2 and $1 - p_1 - p_2$ describe the shares of the three parts of the new mixture. We investigate heterogeneity in real-world data using these extended model in Section 4.5.4.

4.5.2 R Package scModels

We provide the R package **scModels** which contains all functions needed for maximum likelihood estimation (Section 3.3.1) of the considered distribution models. Five applications of the Gillespie algorithm (Gillespie, 1976) allow synthetic data simulation via the basic, switching, bursting, basic-bursting and IG bursting model, respectively (see Sections 4.2, 4.3 and 4.4 for details on the models). Implementations of the likelihood functions for the one-population case and two-population mixtures, with and without zero-inflation, allow inference of the Poisson, NB, PB, DEL and PIG distributions. We provide a new implementation of the PB density – given in Equation (4.9) – , based on our novel implementation of the Kummer function (Definition A.6), also known as the generalized hypergeometric series of Kummer. This became necessary, because the existing R function (`kummerM()` contained in package **fAsianOptions**) was only valid for specific parameter values, and hence, was not suited for optimization in continuous unconstrained space.

4.5.2.1 Implementation of the PB Distribution

We need to evaluate the PMF of the PB distribution (see Example 3.4) in some parts of this thesis. In detail, we calculate the PB distribution in terms of the parameters of the switching model, see Section (4.2.3). The general form of the PMF of the PB$(\alpha, \beta, 0, c)$ distribution for $\alpha, \beta, c > 0$ is given by

$$\mathrm{P}_{\mathrm{PB}}(x|\alpha, \beta, 0, c) = \frac{\Gamma\left(\alpha+\beta\right) c^{x} \Gamma\left(\alpha+x\right)}{\Gamma\left(\alpha\right)\Gamma(x+1)\Gamma\left(\alpha+\beta+x\right)}\ {}_1F_1\left(\alpha+x; \alpha+\beta+x; -c\right)$$

for $x \in \mathbb{N}_0$. To compute this function, the Kummer function ${}_1F_1\left(a; b; z\right)$ (see Definition A.6) needs to be calculated with the following constraints on its parameters:

(i) $z \in \mathbb{R}_{\leq 0}$ (where z is the third parameter of ${}_1F_1$). We only look at the case where $z = -c$ and with Equation (4.9) we set $c = \frac{r_{on}}{r_{deg}} \in \mathbb{R}_{\geq 0}$.

(ii) $a, b \in \mathbb{R}_{\geq 0}$ and $0 \leq a \leq b$. We have $a = \alpha + x$ and $b = \alpha + \beta + x$ and with Equation (4.9) we set $\alpha = \frac{r_{act}}{r_{deg}} \in \mathbb{R}_{\geq 0}$ and $\beta = \frac{r_{deact}}{r_{deg}} \in \mathbb{R}_{\geq 0}$.

Muller (2001) showed how hard it is to compute the Kummer function, because its computational behavior splits into a number of distinct regions, which makes it impossible to have a unified algorithm for all possible input parameters. One of the well-behaved analytical solutions to the function is in the form of an infinite series. Additionally, for specific constraints on the parameters (which are fulfilled when the function appears inside the PB distribution), there exists an integral representation of the solution. Nevertheless, neither the integral nor the infinite sum can be computed directly, and thus approximations and workarounds have to be implemented. There exist different methods to address this problem. On the one hand, there are methods that compute the PB distribution by approximating the integral representation of the Kummer function (see **BPSC** Vu et al., 2016, implemented in Python); while on the other hand methods employ the characteristics of the PB distribution to estimate its parameters, circumventing the evaluation of the Kummer function (see **D3E**, Delmans and Hemberg, 2016, implemented in R). Our approach is to calculate the density by truncating the infinite series solution to the Kummer function at a reasonable error bound. This is also a challenge, because for example the existing R function `kummerM()` (Package:**fAsianOptions**) tries a similar approach, but fails for many parameters. Figure 4.12 displays the comparison between the implementation of the Kummer function by the **fAsianOptions** package and our new implementation.

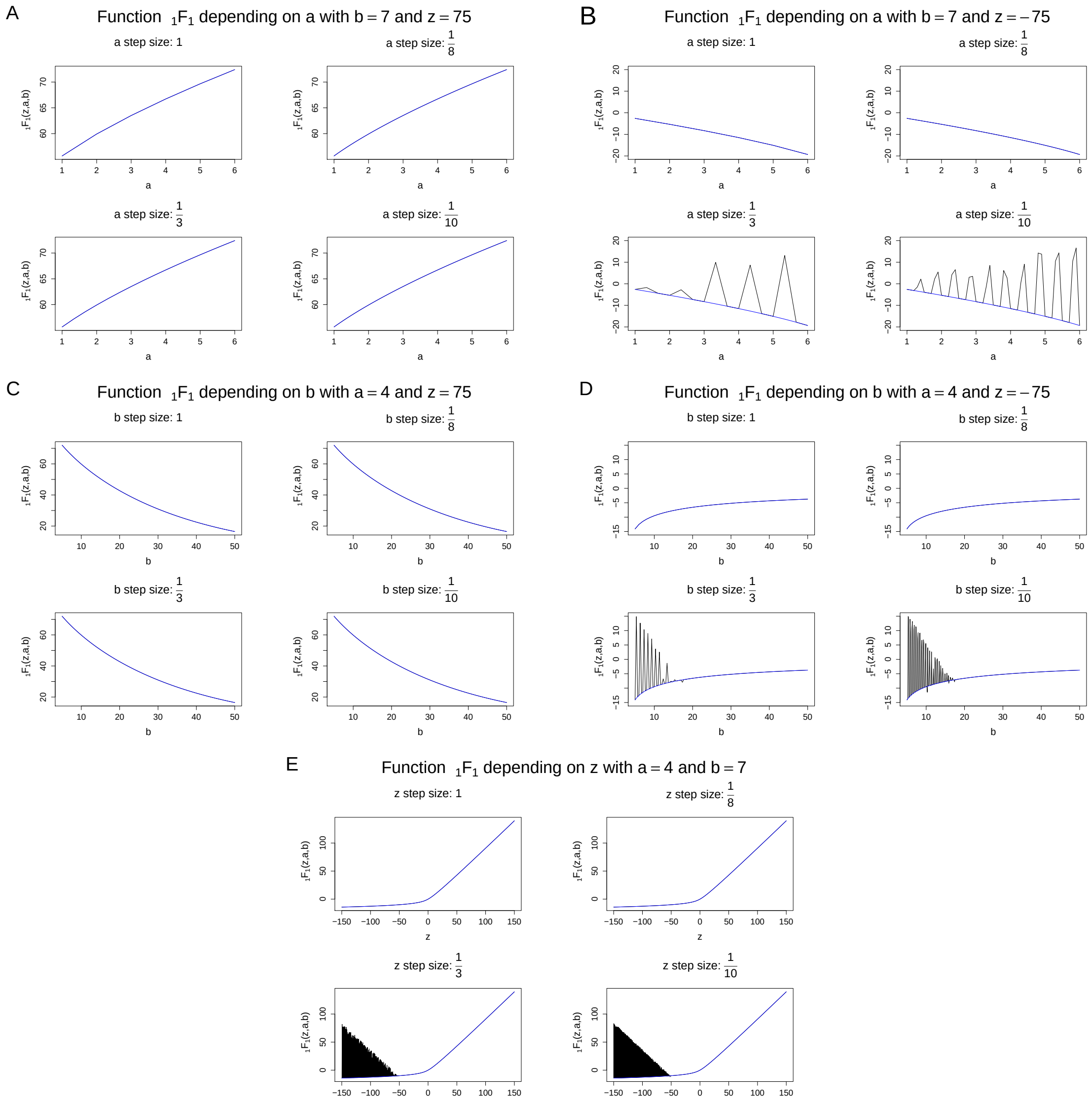


Figure 4.12: Behavior of the Kummer function for different parameter sets based on the implementations of **fAsianOptions** in black and **scModels** in blue. (A,C and E) As long as z is positive, the Kummer function of both packages return the correct values. (B,D and E) As soon as z is negative (smaller than -50) the Kummer function of the **fAsianOptions** returns wrong values for a, b and z for values that cannot be expressed by the general formula $m \cdot 2^{-n}$, $m, n \in \mathbb{N}_0$. This bug is fixed in the new implementation of the Kummer function in **scModels**.

In the following, we will first go into detail of the existing methods and will then present our new implementation. Afterwards we compare our method to existing ones in terms of fitting and computation time.

- **BPSC:** Vu et al. (2016) present how to use the integral representation to calculate the PMF of a PB distribution. This is implemented in their R-package **BPSC**. Vu et al. (2016) define three different beta-Poisson models (they use this name rather than PB) where the so-called three-parameter beta-Poisson model corresponds to the one we presented in the Section 4.2.3 and Equation (4.9), and thus is the one we want to use here and later in the comparison. Parameter estimation is done via likelihood maximization, where two techniques are used to speed up the calculations: First, the authors bin the data into intervals and for each bin the probability is calculated separately via the PMF of the beta-Poisson distribution in this interval. Second, to calculate the PMF of such an interval, the integral-notation of the Kummer function is used and the value of this integral is approximated by using the Gaussian quadrature method. Starting values for α and β for the parameter optimization are calculated based on the method of moments whereas c is assumed to be the maximum of the data points.

- **D3E:** Delmans and Hemberg (2016) implemented two different methods to estimate the parameters of the PB distribution in their **D3E** package that is available in Python: The first implementation is a "fast but inaccurate method"(Delmans and Hemberg, 2016) using the moment matching approach that was first proposed by Peccoud and Ycart (1995). The second implementation is the Bayesian inference method proposed by Kim and Marioni (2013) where gamma priors are used for the parameters α, β and c and a collapsed Gibbs sampler, using slice sampling, is used for parameter estimation. Additionally, **D3E** provides a differential gene expression test by using a likelihood ratio test. To overcome the problem of calculating the Kummer function, a Monte Carlo method is used that approximates the PMF as average of empirical PMFs of a large number of datasets generated from a PB distribution.

- **scModels:** All functions needed to simulate data or estimate distributions are collected in our R package **scModels** which is published on CRAN (`https://CRAN.R-project.org/package=scModels`). The current working version can be found on Github under `https://github.com/fuchslab/scModels`. Included are the Poisson, the NB, the PIG, the DEL and the new implementation of the PB distribution (probability density function, cumulative distribution function, quantile function and random number generation) together with a necessary new implementation of the Kummer function (also called confluent hypergeometric function of the first kind). Additionally, five implemented Gillespie algorithms allow synthetic data simulation via the basic, switching, bursting, basic-bursting and IG bursting mRNA generating process, respectively. Lastly, we added likelihood functions for one population and two population mixtures – with and without zero-inflation – that allow estimation of the Poisson, NB, PIG DEL and the PB distribution. These can be performed with one included wrapper function `fit_params()` that uses the general-purpose optimization function `optim()`.
  As stated above, we implemented a new version in R of the Kummer function that uses the infinite sum representation. The only existing (at least to our knowledge) implementation in R, `kummerM()`, which is contained in the package **fAsianOptions**, works only for some specific parameter choices but not for

others, e. g. for negative z the `kummerM()` does not return the correct values (see Figure 4.12). More specifically, this implementation returns the correct result only for parameter values that can be written as $m\frac{1}{2^n}$ with $m, n \in \mathbb{N}_0$. Because this is impracticable when numerically determining parameters during likelihood optimization, we decided to solve this issue by reimplementing the Kummer function.

Our new implementation aims to be as close as possible to the true solution for the parameter values we need, when the Kummer function is used during calculation of PMF of the PB distribution. Muller (2001) stated that if neither a nor b are negative integers, then the series converges for all finite z. In reality, however, calculations fail when, for example, a and z have opposite signs. The problem arises because of cancellations. One of Kummer's transformations (given in Equation 13.2.39 in Olver et al., 2019) promises to circumvent this problem: Suppose that $a, b \in \mathbb{R}^0_+$ and $0 \leq a \leq b$ but $z \in \mathbb{R}_-$, then

$$M(a, b, z) = \exp(z) M(\tilde{a}, \tilde{b}, \tilde{z}),$$

where $\tilde{a} = b - a, \tilde{b} = b, \tilde{z} = -z$. Now for the new parameters it holds that

(i) $\tilde{z} \in \mathbb{R}_{\geq 0}$.

(ii) $\tilde{a}, \tilde{b} \in \mathbb{R}_{\geq 0}$ for $0 \leq \tilde{a} \leq \tilde{b}$.

With these new constraints, the power series does not have convergence issues. However, the series is difficult to compute due to the limits of machine precision. Consequently, we use the MPFR library (see `https://www.mpfr.org`) for arbitrary-precision floating-point computation. To make the code more readable, we use another MPFR C++ wrapper (`http://www.holoborodko.com/pavel/mpfr/`), written by Pavel Holoborodko. The precision of the temporary results in an expression is chosen as the maximum precision of its arguments, and the final result is rounded to the precision of the target variable.

Although the final result of the function is quite large, the logarithmic value can be casted into double, which is then used further. We implement the iterative algorithm described as Method 1 in Muller (2001). Convergence and error analysis for Taylor series summation using multiple precision arithmetic has been explained in Brent (2010).

Convergence of the Kummer series as given in Definition A.6 can be checked using the ratio test, and an appropriate lower bound on the number of terms needed for computation can be subsequently calculated. One has

$$M(a, b, z) = \sum_{i=0}^{\infty} T_i, \text{ where } T_i = \frac{(a)^i}{(b)^i} \frac{z^i}{i!}.$$

For convergence, we need

$$1 > \lim_{i \to \infty} \left| \frac{T_{i+1}}{T_i} \right| = \lim_{i \to \infty} \frac{(a+i)z}{(b+i)i} ,$$

which is easily fulfilled for all reasonable positive values of a, b, z. With this, we can have a lower bound on the number of terms needed for a good approximation. Specifically, we need to sum up at least until the term where the ratio falls below one. Hence, the condition is

$$\frac{(a+i)z}{(b+i)i} < 1$$

and this implies

$$i^2 + i(b-z) - az > 0.$$

Since only positive values of i are sensible, we have

$$i = \frac{-(b-z) + \sqrt{(b-z)^2 + 4az}}{2} \leq \sqrt{az}\ .$$

Therefore, the series converges after $\sqrt{az}$ terms. Nevertheless, our new implementation of the Kummer function that is contained in **scModels** stops the calculations of the infinite sum as soon as a new summand is smaller than 10^{-6}.

In a simulation study, we compare the implemented functions of the PB distribution that are contained in the three described packages. We first generate sample data on which to test the three packages by using our `gmRNA_switch()` function contained in **scModels**. We use this function to generate data from the switching model as this is the mechanistic model that leads to the PB distribution in steady state (see Section 4.2.3). We simulate 1,000 data points from four different sets of parameters, respectively. Table 4.2 shows the chosen PB parameters which are calculated from the parameters used in the data simulation, $\alpha = r_{act}/r_{deg}$, $\beta = r_{deact}/r_{deg}$ and $c = r_{on}/r_{deg}$, as well as the results of this comparison study. These results are also depicted in Figure 4.13. The estimation procedures and time measurements were performed on a cluster of machines with the following specifications: Intel(R) Xeon(R) CPU E5620 (2.40GHz). Jobs were submitted using the Univa Grid Engine queuing system with 1 GB of memory for each job. Package-specific details of the procedure are described in the following:

- **BPSC:** The function `getInitParam()` estimates initial parameters of the distribution to be passed to the optimization function. The `estimateBP()` function calls the standard `optim()` routine to generate final results.

- **D3E:** D3E is designed for identifying differentially expressed genes based on scRNA-seq data. The data needs to be provided in a tab-separated read-count table, where rows correspond to genes, and columns correspond to cell types. Since it works for differentially expressed genes, the columns in the read-count table have to be labeled for the two different types of cells or conditions. The output is the parameter values of the PB distribution along with other statistics for comparison. Here, we do not aim to test for differential expression but only intend to estimate model parameters for one type of cells. Hence, we have to circumvent this procedure: We use the function `getParamsBayesian()` from inside the package to bypass the differential expression step.

- **scModels:** We use the method of moments combined with bootstrap to predict initial values for the optimization. The final result is obtained by minimizing the negative log-likelihood function that employs the implemented density function `dpb()` of the PB distribution.

The estimation results, given in Table 4.2 and depicted in Figure 4.13 show that all three packages are able to estimate the density function that describes the data well and is close to the true density curve. The obtained values of the negative log-likelihood are in the same range, with our package **scModels** always leading to the lowest or equally low (i. e. best) value. Computing times and parameter estimates are variable and do not show a clear picture besides that the **BPSC** package takes the least amount of time and **scModels** sometimes takes very long. Note that although the values of some of the estimates are far from the true one, the overall likelihood is close. Having a look at the mean and the variance of the PB distribution (see Definition 4.5.2.1) individual parameters might not be identifiable in all cases.

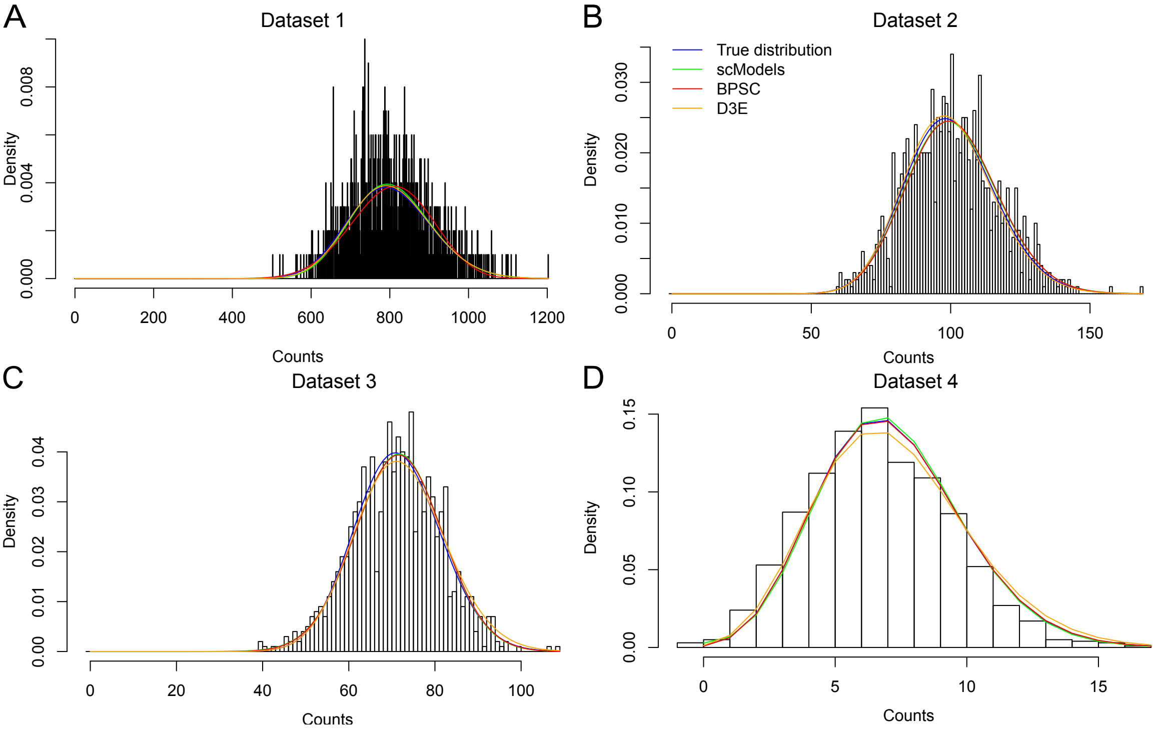


Figure 4.13: Histograms of the four simulated datasets (A-D) and PB densities using the true and estimated parameters from Table 4.2, respectively: true (blue), **scModels** (green), **BPSC** (red) and **D3E** (orange).

	α	β	c	computing time in sec	value of negative log-likelihood
<u>**Dataset 1**</u>					
true values	50	200	4,000	-	6,041
BPSC estimate	23	13	1,243	0.61	6,058
D3E estimate	64	270	4,214	188.77	6,044
scModels estimate	66	2,927	36,384	116,760.21	6,038
<u>**Dataset 2**</u>					
true values	50	200	500	-	4,210
BPSC estimate	41	83	304	1.05	4,208
D3E estimate	62	1,298	2,195	165.67	4,211
scModels estimate	45	135	399	1,528.39	4,208
<u>**Dataset 3**</u>					
true values	50	20	100	-	3,738
BPSC estimate	19	3	82	0.737	3,735
D3E estimate	92	191	221	174.49	3,741
scModels estimate	17	2	80	110.57	3,735
<u>**Dataset 4**</u>					
true values	50	20	10	-	2,415
BPSC estimate	73	69	14	0.686	2,415
D3E estimate	43	2,160	368	163.43	2,418
scModels estimate	0.56	0.0037	7.18	89.67	2,413

Table 4.2: Results of parameter estimation for the PB distribution using the software packages **BPSC**, **D3E** and **scModels**. We simulated four datasets of size 1,000 each. The table shows values of the parameters α, β and c: the true values used for synthetic data generation, and the point estimates obtained through application of the different packages. The last two columns show the computation time measured in seconds for each algorithm and the value of the negative log-likelihood function (computed using the function `scModels::nlogL_pb()` for all) evaluated at the respective parameter values. Smaller values of the negative log-likelihood indicate better point estimates.

With our new implementation of the PB density we did not overcome the problem of time-consuming calculation, but we for the first time provided an implementation of the Kummer function in R valid for all values required inside the PB density.

4.5.3 Simulation Study

In a simulation study, we generate in silico data from the five considered mechanistic models: the basic model (Figure 4.1), the switching model (Figure 4.3A), the bursting model (Figure 4.6), the basic-bursting model (Figure 4.8) and the IG bursting model (Figure 4.11) using our Gillespie implementations in **scModels**. In order to choose realistic values for the rate parameters, we are guided by experimental studies which aim to determine rates of the switching process in specific cases. For example, Suter et al. (2011) identify rates for so-called short-lived genes where mRNA and protein pulses are directly connected to one single on-and-off-switch of a gene. From these we calculate ranges for the basic and the bursting models to ensure that the simulated data is comparable between the models: $r_{tran} = r_{on} \cup r_{act}$ (this is informal notation for the union of the two ranges of r_{on} and r_{act}), $s_{burst} = r_{on}/r_{deact}$ and $r_{burst} = r_{act}$. For the basic-bursting model, we use the same ranges as for the stand-alone basic and bursting models. For the IG bursting model we take for μ_R the same range as r_{tran} and r_{burst} to contain the range of r_{burst} parameter of the bursting model but raise it up to 3 so that the mean burst size of the IG bursting model which is defined by μ_R/r_{burst} lies in the range of s_{burst} of the bursting model. For each of the five considered models, we generate a grid of 1,000 unique parameter sets and generate one dataset for each of those parameter sets resulting in the generation of 5,000 datasets. Each of those contains 1,000 observations. The employed ranges for the parameter grids are displayed in Table 4.3.

Mechanistic model	Rate parameter	Minimum value	Maximum value
Basic model	r_{tran}	0.005	2.5
	r_{deg}	0.001	0.05
Bursting model	r_{burst}	0.005	0.06
	s_{burst}	0.8	250
	r_{deg}	0.001	0.05
Switching model	r_{act}	0.005	0.06
	r_{deact}	0.01	0.6
	r_{on}	0.5	2.5
	r_{deg}	0.001	0.05
Basic-bursting model	r_{tran}	0.005	2.5
	r_{burst}	0.005	0.06
	s_{burst}	0.8	250
	r_{deg}	0.001	0.05
IG bursting model	μ_R	0.005	2.5
	r_{burst}	0.005	3
	r_{deg}	0.001	0.05

Table 4.3: Ranges of rates in the simulation study for all five different transcription models.

As a proof of concept, we estimate the five corresponding distributions, i. e. the Poisson, PB, NB, PIG, and the DEL distribution, on all generated datasets via maximum likelihood estimation (Section 3.3.1).

As described in Section 3.4 we use the BIC to select the model which fits the data best. Afterwards we perform the χ^2-test to assess the goodness-of-fit (GOF), i. e. whether the distribution estimated with the model fits to the underlying data. We neglect those models respectively datasets for which the estimated distribution is rejected at the 5% significance level. This reduces the total number of 1,000 simulated datasets per model to the amounts displayed in Figure 4.14A.

We investigate whether the selected distributions correspond to the distributions that arise from the respective mechanistic models: For the datasets generated from the basic model, model selection via BIC (after GOF) indeed prefers the Poisson distribution in most cases, independently of the used distribution parameter λ

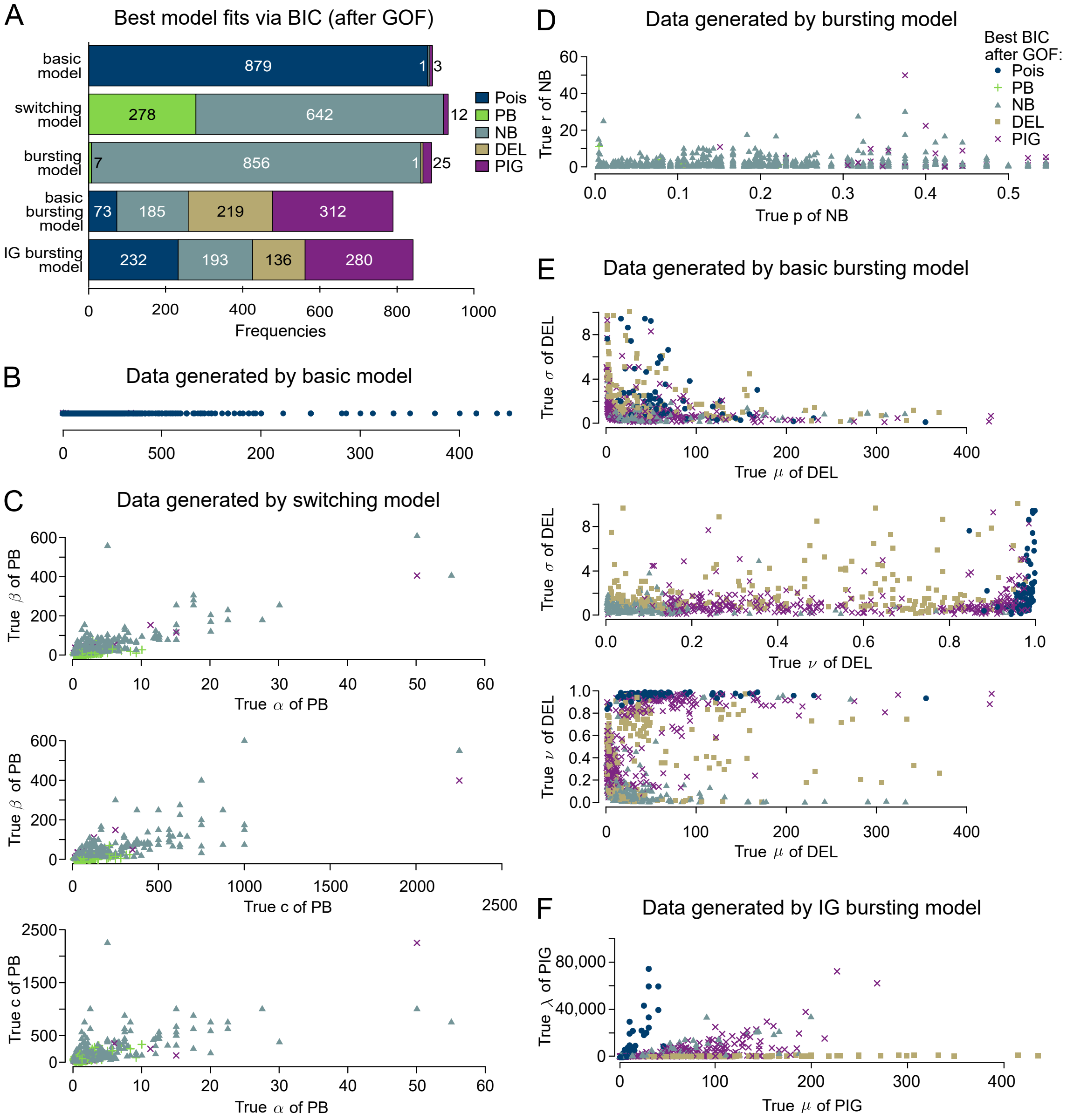


Figure 4.14: Model selection on in silico data: (A) Frequencies of chosen distributions (Poisson, PB, NB, DEL, PIG) via BIC (after GOF) based on datasets generated by the five different transcription models (basic, bursting, switching, basic-bursting, IG bursting) using the Gillespie algorithm. (B-F) Employed parameter values (indicated by horizontal/vertical position) and chosen distributions (indicated by color/symbol) for basic model (B), switching model (C), bursting model (D), basic-bursting model (E) and IG bursting model (F). The names of the parameters correspond to those in Definitions A.7, A.8, Equation (3.1), Definition A.10 and Example 3.5.

(Figure 4.14A, first row, and Figure 4.14B). In contrast, for datasets generated by the switching model, selection via BIC (after GOF) mostly chooses either the NB or the PB distribution (Figure 4.14A, second row). The choice seems to depend on the employed rate parameters: Figure 4.14C indicates a tendency towards the PB distribution for low values of β; otherwise, the NB distribution often seems to model the data generated by the switching model sufficiently well. This indicates that, the NB distribution is complex enough to describe the data generated from the switching model. The BIC decides in many cases that a potentially better fit is not worth the extra effort for estimating an additional parameter in the PB distribution. For datasets generated by the bursting model, BIC (after GOF) selection picks the NB distribution for the majority of the time without any obvious bias (Figure 4.14A third row and Figure 4.14D). Data generated either via the basic-bursting model or the IG bursting model do not show a general distribution preference. The last two bars in Figure 4.14A show that all distributions but the PB are selected in large numbers. Figure 4.14E shows that for large ν parameter of the DEL distribution the data is best modeled by a Poisson distribution and for small ν with simultaneously small σ the NB is preferred. This is what we expect, since the basic-bursting model generates in theory data following a DEL distribution, which is the sum of a NB and a Poisson distribution. The parameter $\nu = \frac{\lambda}{\lambda + \frac{r(1-p)}{p}}$ describes the relations of the means of the involved Poisson and NB distributions. If $\nu \approx 1$, the mean of the Poisson distribution is much higher than the mean of the NB. Therefore the Poisson dominates the distribution and therefore the data can be modeled sufficiently well by a Poisson distribution. The other way around, if $\nu \approx 0$, the NB dominates the DEL distribution and therefore data can be modeled sufficiently well by a single NB distribution. We cannot identify parameter ranges where the model selection prefers a PIG over a DEL. Lastly data generated by the IG bursting model is best fitted by a Poisson distribution for small μ and additional large λ. In this case μ_R is much smaller than r_{burst} which induces less IG-transcription and more bursts. At the same time burst sizes are very small so that this process can be approximated by the basic process and therefore the Poisson distribution is sufficient to model the data. On the other side, the DEL distribution seems to model the data best for large μ and small λ which corresponds to a process containing mainly the IG-transcription and very rare bursts with large burst sizes. We do not know anything about such a pure IG-process but apparently it is best modeled by a more complex distribution such as the DEL. We cannot find a criterion when to use the NB or the PIG distribution to model the data. Clearly, the PIG is the distribution most often preferred. Both have the same complexity in terms of parameter numbers. Note, that the IG bursting model does not have the bursting model as a special case: Choosing μ_R small so that IG-transcription does not happen, we note that the burst sizes also depends on μ_R. This means that for a mean burst size greater than 1, r_{burst} needs to be smaller than μ_R. This in turn means that bursts occur less frequently than the IG-transcription. To better understand the five univariate distributions used above, we depict in Figure 4.15 how the shapes of the distributions differ for the same mean E and variance Var.

Four mean-variance combinations are selected. With small and large mean value and, based on this, small and large variance.

The parameters of the five distributions in dependence of mean and variance are easily computed for the two parametric distributions NB and PIG. NB can be directly parameterized via its mean $\mu = E$ and size $\alpha = \frac{\mu^2}{\mathrm{Var}-\mu}$ (see Definition A.8). The PIG is parameterized via mean $\mu = E$ and $\sigma = \frac{\mathrm{Var}-\mu}{\mu^2}$ (see Example 3.5).

Since the PB and the DEL distributions are defined by three parameters, one parameter needs to be chosen manually. Therefore we discuss several choices to show the whole flexibility of these distributions. In detail for the PB, we select values for c. Note that c has to be larger than the mean we set it to be either 10E or 1000E. With this

$$\alpha = \frac{\left(\frac{\mathrm{E}^2(c-\mathrm{E})}{\mathrm{Var}-\mathrm{E}} - \mathrm{E}\right)}{c} \quad \text{and} \quad \beta = \frac{\left(\frac{\mathrm{E}^2(c-\mathrm{E})}{\mathrm{Var}-\mathrm{E}} - \mathrm{E}\right)(c-\mathrm{E})}{\mathrm{E}\,c}$$

can be calculated.

For the DEL distribution we set the ν parameter manually. It needs to be between 0 and 1 and therefore we selected it to be 0.1, 0.5 and 0.9, respectively. Since the DEL distribution is also parameterized via the mean $\mu = E$ we only need to calculate $\sigma = \frac{\mathrm{Var}-\mu}{\mu^2(1-\nu)^2}$.

Figure 4.15 shows that the NB distribution and the PB distribution with the large c always lie on top of each other. This explains why often the NB distribution is good enough to model data that originate from the switching model or a PB distribution.

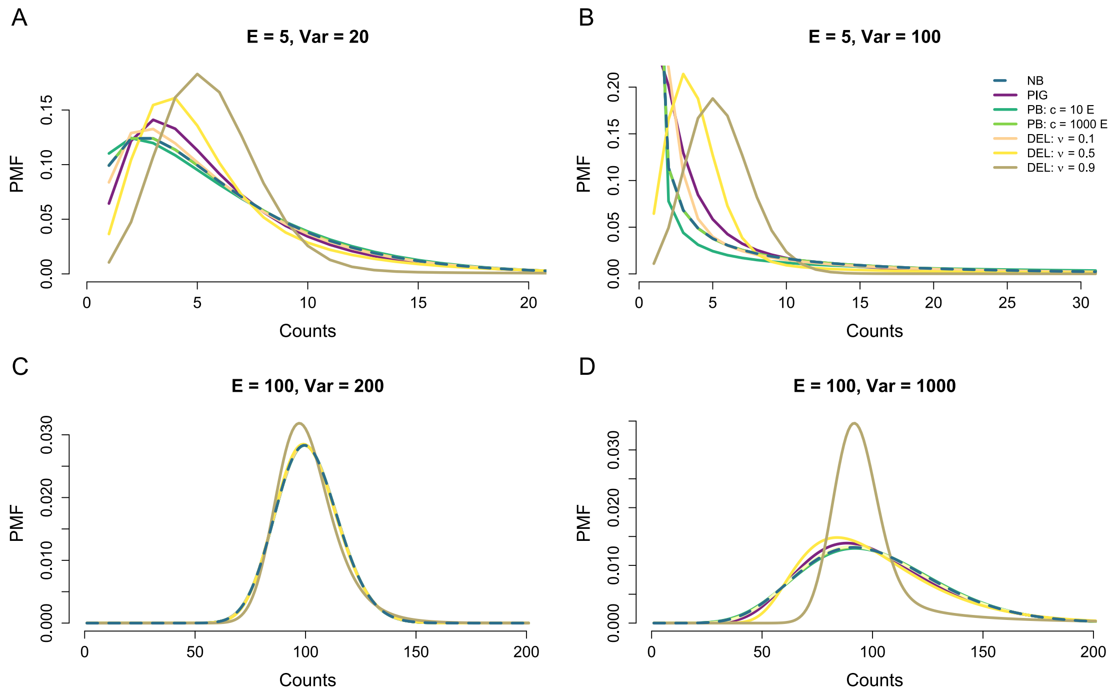
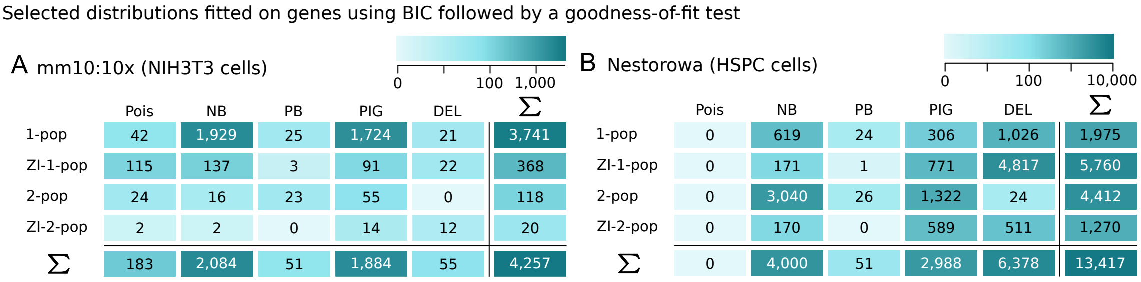


Figure 4.15: Comparison of the characteristics of the four distributions for four different mean and variance settings. PB and DEL are defined by three parameters and hence this third parameter is chosen several times to show the bandwidth of shapes of possible distributions.

Furthermore the two PB do not show very different shapes. Compared to these, the PIG distribution differs in its shapes. However, the DEL distributions, especially the shape of the one with the largest ν of 0.9, is very different to all other distributions. Since a large ν means that the distribution resembles a Poisson distribution (not shown here), this seems reasonable. Taken together since the five distributions are able to model very similar shapes, often the less complex NB or PIG distribution are selected. These two share the same complexity and therefore often only very small differences in BIC make the final decision.

4.5.4 Application to Real-World Data

We perform a comprehensive comparison of the considered mRNA count distributions, that is the Poisson, NB, PB, DEL and PIG distribution, when applied to real-world data. Within each of the five distributions we further consider mixtures of two populations (from identical distribution types but with different parameters) with and without additional zero-inflation, as described in Section 4.5.1. In total, we investigate twenty different models as shown in Figure 4.16. The numbers of parameters in these models are listed in Table 4.4.



Selected distributions fitted on genes using BIC followed by a goodness-of-fit test

A mm10:10x (NIH3T3 cells)

	Pois	NB	PB	PIG	DEL	Σ
1-pop	42	1,929	25	1,724	21	3,741
ZI-1-pop	115	137	3	91	22	368
2-pop	24	16	23	55	0	118
ZI-2-pop	2	2	0	14	12	20
Σ	183	2,084	51	1,884	55	4,257

B Nestorowa (HSPC cells)

	Pois	NB	PB	PIG	DEL	Σ
1-pop	0	619	24	306	1,026	1,975
ZI-1-pop	0	171	1	771	4,817	5,760
2-pop	0	3,040	26	1,322	24	4,412
ZI-2-pop	0	170	0	589	511	1,270
Σ	0	4,000	51	2,988	6,378	13,417

Figure 4.16: Frequencies of chosen distributions via BIC followed by a χ^2 GOF test across genes of two real-world datasets: (A) 4,257 genes measured in 3,356 homogeneous NIH3T3 mouse cells (see Official 10x Genomics Support, 2017) and (B) 13,417 genes measured in 1,656 heterogeneous mouse HSPCs (see Nestorowa et al., 2016). Twenty different distributions were fitted to the genes: Poisson (Pois), negative binomial (NB), Poisson-beta (PB), Poisson-inverse Gaussian (PIG) and Delaporte (DEL) distributions. Each was fitted as a univariate (1-pop) without and with zero-inflation (ZI-1-pop) and as a mixture of two distributions (2-pop) without and with zero-inflation (ZI-2-pop).

We estimate these twenty distribution models on two real-world datasets. Before interpreting the results we will explain the data preparation. The first one contains 3,356 homogeneous NIH3T3 mouse cells (a standard cell line established from primary mouse embryonic fibroblast cells) and has been generated using the 10x Chromium technique, see details in Section 2.2.4 which incorporates UMIs. This is a procedure that collapses all reads that are measured from the same mRNA transcript to one molecule count (see Section 2.2.2 and Islam et al., 2014). It is part of the publicly available 10x dataset "6k 1:1 Mixture of Fresh Frozen Human (HEK293T) and Mouse (NIH3T3) Cells" (Official 10x Genomics Support, 2017). Here, we refer to this dataset as mm10:10x.

The second dataset stems from Nestorowa et al. (2016), contains 1,656 mouse HSPCs (Hemapoietic stem and progenitor cells) and was generated using the Smart-Seq2 (Picelli et al., 2014) protocol, and thus does not employ UMIs.
The data preprocessing including a gene filtering has been performed as follows.

- **mm10:10x dataset** (Official 10x Genomics Support, 2017). This dataset contains UMI counts. The raw UMI matrix (only the mouse part) consists of 27,998 genes in 3,427 cells. To filter out cells with only a few expressed genes that could, for example, be generated by empty droplets, we apply a cell filter that only selects cells that express more than 1,500 genes. The gene filter is slightly less strict than the one for the second dataset as UMI count matrices show smaller entries (by definition several read counts collapse to less UMI counts). Thus, we filtered for genes that are expressed in at minimum ten cells with at minimum three UMIs.

- **Nestorowa dataset** (Nestorowa et al., 2016). As described above, this data was generated using the Smart-Seq2 protocol and thus the resulting data consists of read counts. The original data matrix contains 45,771 genes and 1,656 cells. We use two filters: The first one selects only those genes that have mean expression larger than one, whereas the second filter additionally removes all genes that are only lowly expressed, i.e. after application of this filter, only those genes remain that have at minimum five reads in at minimum 20 cells. After having applied the two filters, we are left with a read count matrix of 16,364 genes and 1,656 cells.

Using these gene matrices, we estimate the model parameters of the twenty considered models via maximum likelihood estimation, and performed model selection analogously as described in the simulation study in Section 4.5.3. In contrast to the simulation study, the model extensions for heterogeneity and dropout are included. Figure 4.16 summarizes the frequencies of the chosen models across genes. We only display those choices where the chosen distribution (via BIC) with estimated parameters was not rejected by a GOF χ^2-test at 5% significance level with multiple testing correction (see Section 3.4).
Figure 4.16A shows that in the mm10:10x data 4,257 genes remained after filtering and the GOF test. For 49% of these genes, a NB distribution variant and for 44% of these genes, a PIG distribution variant was chosen as most appropriate model. However, a standard distribution (for one population, without zero-inflation) was sufficient in the majority of cases.
We looked for commonalities between the gene profiles that led to the same distribution choice in the mm10:10x dataset. To that end, we estimated one-population models of the Poisson, NB, PB, PIG and DEL distributions for all genes and chose the most appropriate model among those five based on BIC and GOF. Figure 4.17 visualizes the values of the parameter estimates for each model and indicate the chosen models by different colors. Figure 4.17B shows for example that if the NB

distribution is estimated, the following pattern can be observed: If the NB distribution is also the chosen one, the corresponding estimated parameters cover wide ranges $p \in (0, 1)$ and $r \in [0, 12]$. Same holds for the PIG distribution. In contrast, gene profiles that are most adequately described by a Poisson distribution would have resulted in a fairly large value of the parameter p in the NB distribution (i. e. $p > 0.2$, but more than 90% of them show $p > 0.6$) and larger values of r (i. e. $r \in [0, 16]$). Those genes that chose the PB or DEL distribution would have had smaller values in both parameters, namely $p < 0.6$ and $r < 7$. Similar observations can be made for the other distributions. Therefore, Figure 4.17 suggests an interdependence between the chosen one-population distributions and the range of the parameter estimates.

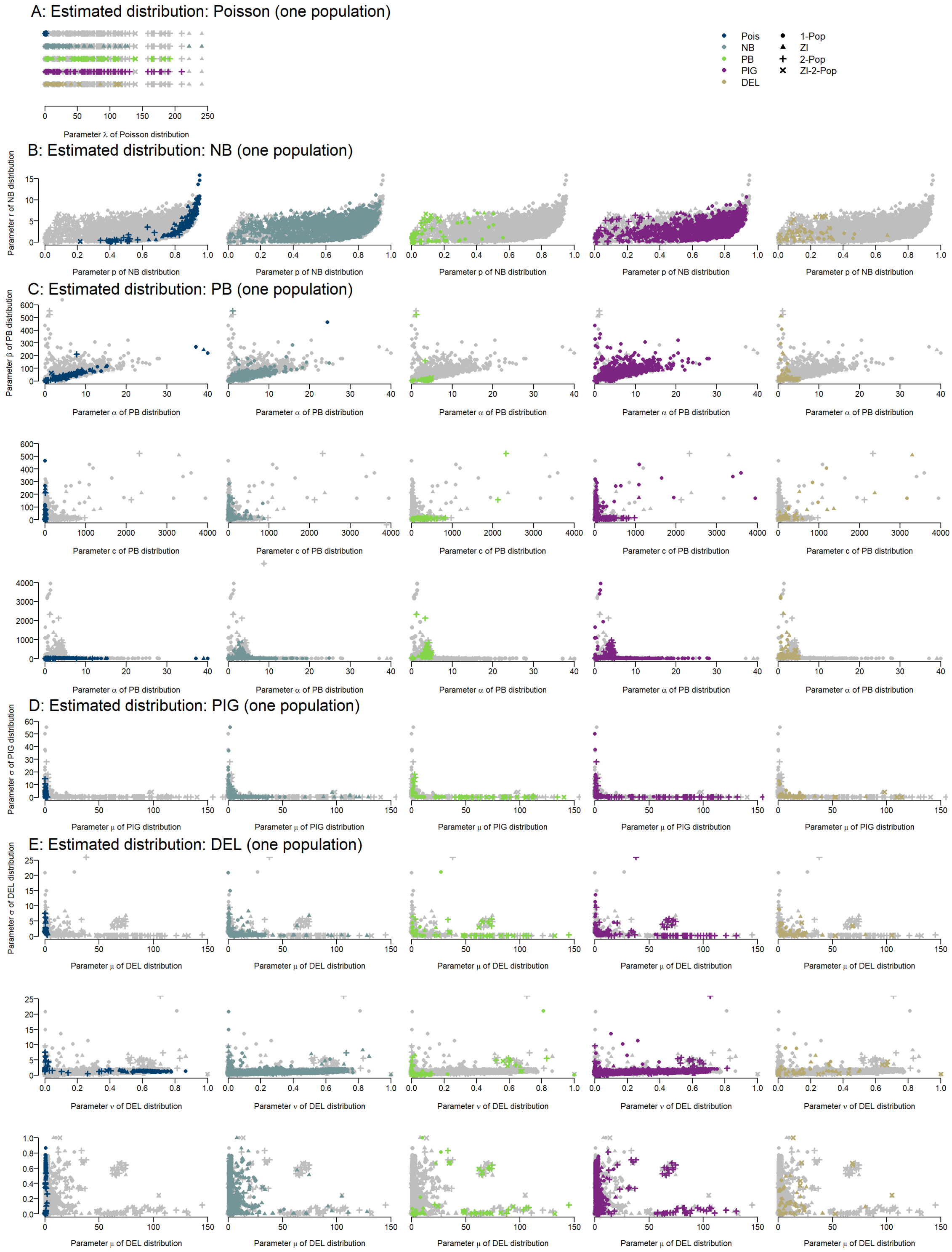


Figure 4.17: We estimated one-population models of the Poisson, NB, PB, PIG and DEL distributions for all genes in the mm10:10x dataset and plot their fitted parameters for each and color the most appropriate model based on BIC after GOF. (A) Estimated λ parameters for the Poisson distribution. Each dot corresponds to one gene. In the first column, estimated values are colored in dark blue for those genes where the Poisson distribution was chosen. In the second column, turquoise symbols indicate estimated values in the Poisson model where the NB distribution would have been preferred. In the third column, green color indicates the estimates for those genes that chose the PB distribution. The forth column shows the genes which prefer the PIG distribution in violet and the last line shows in yellow the Poisson parameters of those genes that were best fitted by a DEL distribution. (B-E) Similarly for the NB, PB, PIG and DEL distributions.

In Figure 4.16A, we observed a relatively large number of genes (in comparison to Figure 4.16B) for which mRNA count data from the mm10:10x dataset (Official 10x Genomics Support, 2017) was best described by some variant of the Poisson distribution, a distribution model that—for general contexts—is considered too simple. We thus searched for patterns in the gene ontology (GO) terms of these genes, see Figure 4.18, but did not observe any apparent differences in the characteristics of the Poisson genes (i. e., those genes where the Poisson distribution was chosen) and the non-Poisson genes. To conduct this analysis, we used GO term information from `http://supfam.org/SUPERFAMILY/cgi-bin/go.cgi` and the R packages **biomaRt** and **GOfuncR**. **biomaRt** determines all GO terms of a gene, and **GOfuncR** determines all parents of a GO term. This information was then filtered for the first children GO terms.

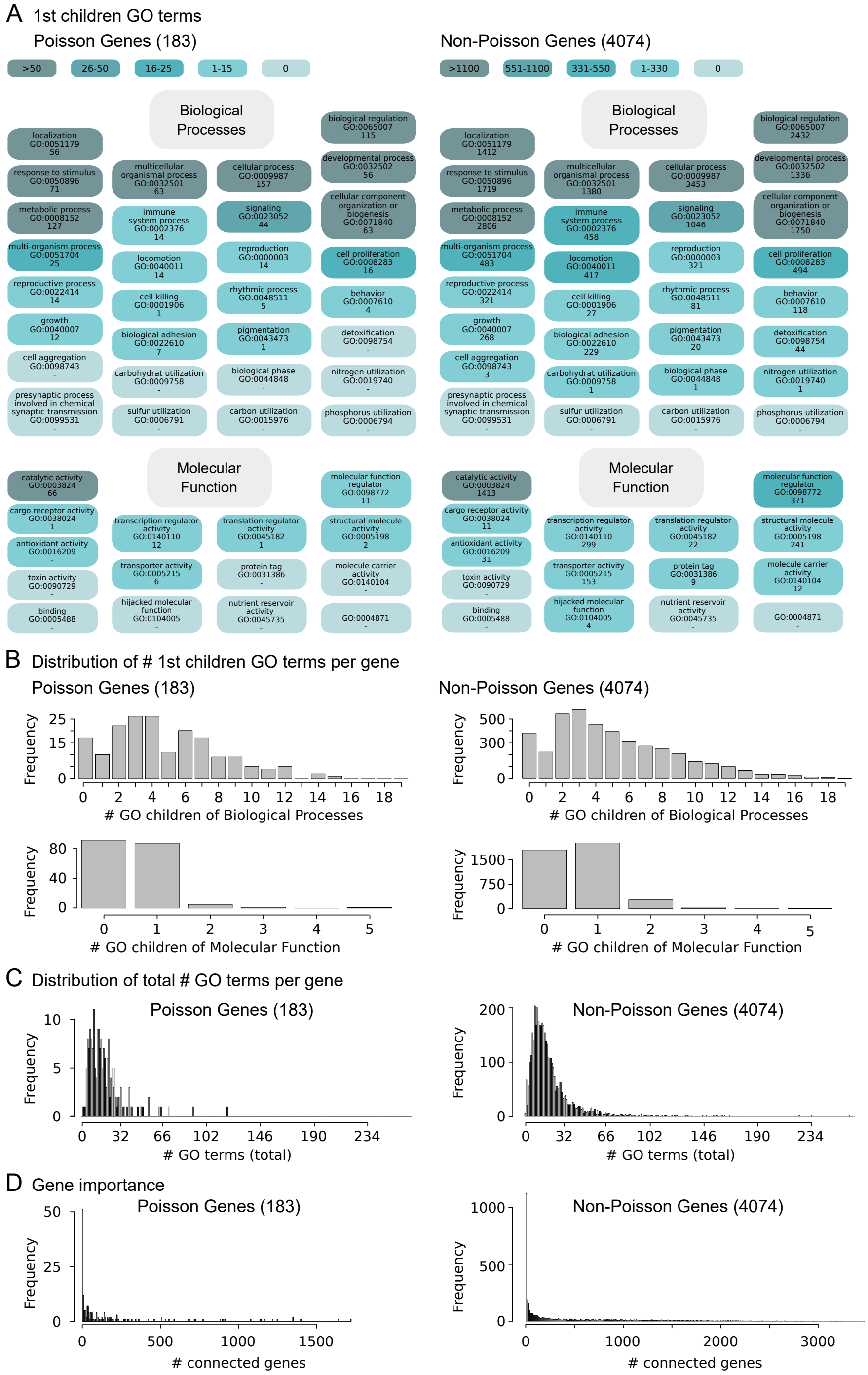


Figure 4.18: GO term analysis of the mm10:10x dataset (Official 10x Genomics Support, 2017) comparing groups of genes which where best described by a variant of the Poisson model and those that were not (see Figure 4.16A). (A) Amount of Poisson and non-Poisson genes (after GOF) that are contained in the first level of GO term children of the families *biological process* and *molecular function*. (B) Distribution of the first children GO terms of the families *biological process* and *molecular function* for Poisson and non-Poisson genes. (C) Distribution of the overall number of GO terms a gene is contained in. GO terms were taken from the initial **biomaRt** determination. (D) Gene importance of Poisson and non-Poisson genes: Functional coupling network of genes taken from `funcoup.sbc.su.se`. Each link with weight > 0.75 was taken and the distribution of the number of coupled genes per gene in this network is plotted.

In the Nestorowa et al. (2016) data, 16,364 genes remained after filtering, of which 13,417 were not rejected by the GOF test. Figure 4.16B shows that nearly half of the genes prefer some variant of the DEL distribution. Most remaining genes prefer either some variant of the NB distribution (30%) or PIG distribution (22%). There, however, most often the mixture of two distributions best describes the mRNA counts. This pattern can be explained by taking a closer look at the gene expression counts of the affected genes.

In Figure 4.19, we exemplarily display the count frequencies for five known blood differentiation genes from this dataset (see Paul et al., 2015). Most of those genes not only show many zeros, but also many low non-zero counts, i. e. many ones, twos etc., next to higher counts. Such expression profiles are not covered by a simple zero-inflated model but prefer a mix of two distributions, one of them mapping to low expression values.

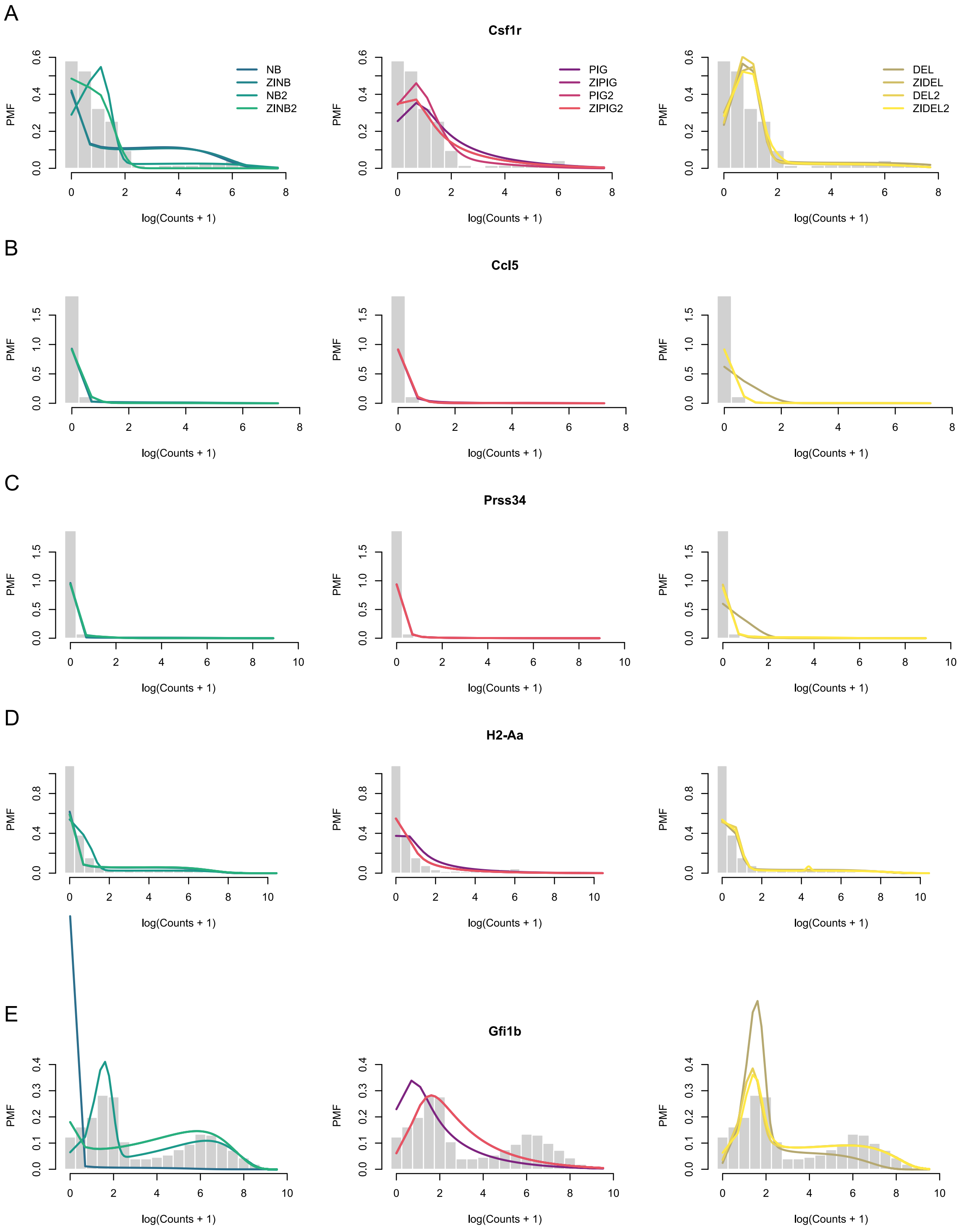


Figure 4.19: (A)-(E) Log-transformed mRNA count histograms for five genes (based on 1,656 single cells) from the dataset by Nestorowa et al. (2016), known as blood differentiation marker genes (see Paul et al., 2015). Colored lines indicate the densities of the estimated distribution variants: NB distributions (left: shades of turquoise), PIG distributions (middle: shades of violet) and DEL distributions (right: shades of yellow).

Table 4.4 contains more estimation details including BIC values for these five genes.

Model – abbr. (# parameter)					Csf1r	Ccl5	Prss34	H2-Aa	Gfi1b
BIC	Pois	1-pop	– Pois	(1)	377,837	62,721	194,047	1,224,382	1,690,010
		ZI-1-pop	– ZIPois	(2)	330,413	29,741	107,502	984,306	1,618,095
		2-pop	– Pois2	(3)	60,829	9,930	20,889	390,636	489,696
		ZI-2-pop	– ZIPois2	(4)	74,845	8,759	150,63	568,092	502,270
	NB	1-pop	– NB	(2)	10,295	1,878	1,545	8,454	29,842
		ZI-1-pop	– ZINB	(3)	10,292	1,865	1,490	8,454	18,653
		2-pop	– NB2	(5)	**8,505**	1,693	1,387	**7,672**	**17,585**
		ZI-2-pop	– ZINB2	(6)	9,978	1,713	1,401	8,477	18,676
	PB	1-pop	– PB	(3)	10,407	1,865	1,490	8,516	18,675
		ZI-1-pop	– ZIPB	(5)	10,414	1,897	1,555	8,618	18,803
		2-pop	– PB2	(7)	10,187	1,920	1,570	8,467	18,778
		ZI-2-pop	– ZIPB2	(8)	10,727	1,937	1,653	8,564	18,787
	PIG	1-pop	– PIG	(2)	8,979	1,732	**1,338**	8,040	18,988
		ZI-1-pop	– ZIPIG	(3)	8,888	1,729	1,345	7,817	18,355
		2-pop	– PIG2	(5)	8,798	1,690	1,357	7,832	18,369
		ZI-2-pop	– ZIPIG2	(6)	8,910	1,732	1,368	7,840	18,377
	DEL	1-pop	– DEL	(3)	8,597	8,458	16,810	7,710	18,600
		ZI-1-pop	– ZIDEL	(4)	8,516	**1,689**	1,383	7,697	17,669
		2-pop	– DEL2	(7)	8,571	1,711	3,397	7,719	17,692
		ZI-2-pop	– ZIDEL2	(8)	8,520	1,716	1,600	7,716	17,650
Selected model					NB2	ZIDEL	PIG	NB2	NB2
p-value of GOF (χ^2) test					0.01515	0.5542	0.05752	0.9371	2.233e-04
Percentage of zero counts					29.0%	91.4%	93.5%	54.0%	6.2%
Percentage of one counts					26.3%	5.6%	3.7%	19.1%	8.1%
Percentage of counts larger than one					44.6%	3.0%	2.7%	26.9%	85.7%

Table 4.4: BIC values for selected blood differentiation marker genes (based on 1,656 single cells) of the Nestorowa et al. (2016) dataset. *Columns:* Results for five genes Csf1r, Ccl5, Prss34, H2-Aa, Gfi1b. *Rows:* BIC values for all twenty estimated models; the smallest BIC shows in bold which model is selected. The selected models and the corresponding p-value of the GOF test, together with percentages of zero counts, one counts, and counts larger than one are listed at the bottom of the table. These five genes and the corresponding distribution fits are depicted in Figure 4.19.

Taken together, the zero-inflated DEL distribution and the two population mixture of the PIG or the NB distribution are chosen for most gene profiles. This clearly proves the heterogeneous nature of the data.

4.5.5 NB Distribution as Commonly Chosen Count Model

While the mechanistic models and their steady-state distributions describe actual mRNA contents in single cells, real-world data underlies technical variation, such as measurement errors in addition to biological complexity. In Section 4.5.3, we investigated in a simulation study and in Section 4.5.4 on real-world data which distributions were most appropriate among those considered to describe gene expression profiles in general. The simulation study showed that a NB distribution may be best suited even if the in silico data had been generated from the switching model. Using more complex data generation models, such as the basic-bursting model or the IG bursting model results in a mix of distribution preferences. These more complex models generate data for certain parameters that are very similar to the data of the basic or of the bursting model. A direct comparison of the five investigated distributions showed that all of them can model very similar shapes of distributions and therefore distribution selection is often only made with regard to distribution complexity. Since the NB and PIG distribution share the same complexity in terms of parameter numbers, there is often no substantial difference. The same applies to the PB and DEL distribution, with the difference that calculating the PB distribution is the most time-consuming. We cannot find relevant advantages of the PIG compared to the NB. This implies that the NB distribution is suited to be chosen as the distribution family to model all different types of gene expression profiles, although we know that the mechanistic process might have been more complex in reality. Also in the real-data application, the NB distribution was often chosen. In line with our expectations, gene profiles of the non-UMI-based dataset by Nestorowa et al. (2016) showed strong preference for a two-population mixture or zero-inflated variant of the NB or PIG distribution. Surprisingly the DEL distribution was most often selected for homogeneous genes with or without zero inflation. Apparently its additional parameter and thus higher complexity was sufficient to model the data but still less complex than a two-population model of the less complex NB or PIG distributions. In contrast, the mm10:10x dataset consists by construction of homogeneous cells, and 10x Chromium is not known for large amounts of unexpected zeros in the measurements. Accordingly, the homogeneous NB or PIG distribution was sufficient for most gene profiles here. For 4% of the considered genes in the mm10:10x dataset, mRNA counts were most appropriately described by some form of the Poisson distribution. We have examined these 183 genes for functional similarities; while estimated parameters show some apparent pattern (see Figure 4.17), we did not find any defining biological characteristics (Figure 4.18).

Similar to us, Vieth et al. (2017) performed model selection among Poisson, NB and PB distributions by BIC and GOF on several publicly available datasets. Although they used the method of Vu et al. (2016) it was not possible for them to calculate a GOF statistics for PB fits. In our study, we represent the PB density in terms of the Kummer function, which allows us to compute the GOF statistics accordingly. Furthermore, we added two additional distributions, the PIG and the DEL distribution. With more included distributions we can confirm the tendency towards the NB distribution as preferred distribution that Vieth et al. (2017) observed.

Different sequencing protocols might lead to differences in distributions and also might generate data of different magnitudes. Ziegenhain et al. (2017) applied various sequencing methods to cells of the same kind to understand the impact of the experimental technique on the data. Based on the data generated in that paper, Chen et al. (2018) investigated differences in gene expression profiles between read-based and UMI-based sequencing technologies. They concluded that, other than for read counts, the NB distribution adequately models UMI counts. Townes et al. (2019) suggest to describe UMI counts by multinomial distributions to reflect the nature of the sequencing procedure; for computational reasons, they propose to approximate the multinomial density again by an NB density. Overall, the NB distribution appears sufficiently flexible to hold independently of the specific sequencing approach.

4.6 Discussion and Conclusion

In this chapter, we derived a mechanistic model for stochastic gene expression that results in the NB distribution as steady-state distribution for mRNA content in single cells. According to the so-obtained bursting model, transcription happens in chunks, rather than in a one-by-one production as commonly assumed in mechanistic modeling (Dattani and Barahona, 2017). We discuss the biological plausibility of bursty transcription further below. The consideration of the bursting model and its derivation is interesting from both practical and theoretical points of view:

First of all, the NB distribution is defined through two parameters whereas the PB distribution typically requires three parameters to be specified in the current context. Therefore, the parameters of a NB distribution are computationally less elaborate to estimate, given some data, than the ones of a PB distribution. Several tools employ the NB distribution to parameterize mRNA read counts (see Table 4.1). However, other than for the Poisson and the PB distributions (Figures 4.1 and 4.3), there has been no explicitly described mechanistic transcription model leading to the NB distribution. In Section 4.3.1 we provide an explanatory transcription process. Note that the PIG distribution is of the same complexity than the NB distribution in terms of parameters. But none of the listed tools (see Table 4.1) uses this distribution. Therefore, we keep in mind that the PIG could be another suitable distribution equivalent to the NB but does not play a role in current tool development. When studying simulated and real-world data in Section 4.5, we could not find relevant advantages of using the PIG compared to the NB. Figure 4.20 contains an overview of the five presented models and their assumptions, i. e. the mechanistic process, the subordinator of the OU process that drives transcription and the resulting steady state distribution for the mRNA counts.

Second, we demonstrated how to generally link a probability distribution to an OU process and derive a mechanistic model. This brings a new field of mathematics to single-cell biology. The procedure can be used to deduce possible mechanistic processes leading to different steady-state distributions, exploiting the rich literature on OU processes from financial mathematics.

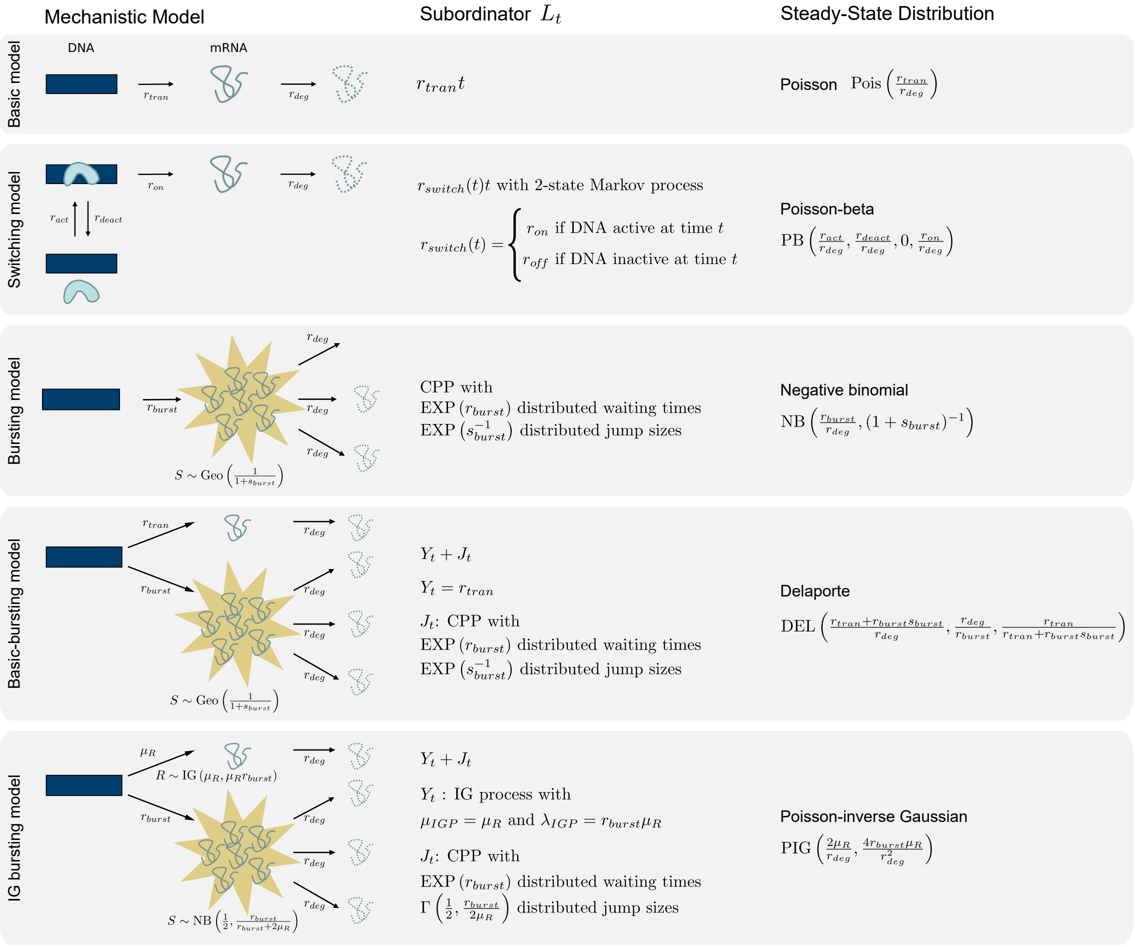

Mechanistic Model
Subordinator L_t
Steady-State Distribution
Basic model
DNA
mRNA
r_{tran}
r_{deg}
$r_{tran}t$
Poisson $\text{Pois}\left(\frac{r_{tran}}{r_{deg}}\right)$
Switching model
r_{on}
r_{deg}
r_{act}
r_{deact}
$r_{switch}(t)t$ with 2-state Markov process
$r_{switch}(t) = \begin{cases} r_{on} \text{ if DNA active at time } t \\ r_{off} \text{ if DNA inactive at time } t \end{cases}$
Poisson-beta
$\text{PB}\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}, 0, \frac{r_{on}}{r_{deg}}\right)$
Bursting model
r_{burst}
r_{deg}
$S \sim \text{Geo}\left(\frac{1}{1+s_{burst}}\right)$
CPP with
EXP (r_{burst}) distributed waiting times
EXP (s_{burst}^{-1}) distributed jump sizes
Negative binomial
$\text{NB}\left(\frac{r_{burst}}{r_{deg}}, (1+s_{burst})^{-1}\right)$
Basic-bursting model
r_{tran}
r_{burst}
r_{deg}
$S \sim \text{Geo}\left(\frac{1}{1+s_{burst}}\right)$
$Y_t + J_t$
$Y_t = r_{tran}$
J_t: CPP with
EXP (r_{burst}) distributed waiting times
EXP (s_{burst}^{-1}) distributed jump sizes
Delaporte
$\text{DEL}\left(\frac{r_{tran}+r_{burst}s_{burst}}{r_{deg}}, \frac{r_{deg}}{r_{burst}}, \frac{r_{tran}}{r_{tran}+r_{burst}s_{burst}}\right)$
IG bursting model
μ_R
$R \sim \text{IG}(\mu_R, \mu_R r_{burst})$
r_{burst}
r_{deg}
$S \sim \text{NB}\left(\frac{1}{2}, \frac{r_{burst}}{r_{burst}+2\mu_R}\right)$
$Y_t + J_t$
Y_t : IG process with
$\mu_{IGP} = \mu_R$ and $\lambda_{IGP} = r_{burst}\mu_R$
J_t: CPP with
EXP (r_{burst}) distributed waiting times
$\Gamma\left(\frac{1}{2}, \frac{r_{burst}}{2\mu_R}\right)$ distributed jump sizes
Poisson-inverse Gaussian
$\text{PIG}\left(\frac{2\mu_R}{r_{deg}}, \frac{4r_{burst}\mu_R}{r_{deg}^2}\right)$

Figure 4.20: Overview of the five transcription and degradation models and their assumptions.

Third, although we focused on the resulting steady-state distributions of the mechanistic models here, our mathematical framework also provides model descriptions in terms of stochastic processes. Nowadays, sequencing counts are commonly available as snapshot data. However, time-resolved measurements may become standard (Golding et al., 2005), and in that case our models open up the statistical toolbox of stochastic processes to extract information from interdependencies within single-cell time series.

Limiting cases of the switching model that give rise to the NB **distribution are biologically unrealistic.** The NB and PB distributions have been linked before. Among others, Raj et al. (2006) and Grün et al. (2014) have shown that the NB distribution is an asymptotic result of the switching model and the corresponding PB distribution as shown in Section 4.2.3.1. However, this result holds only under biologically unrealistic assumptions as we elaborate in the following. Our derivation of the NB steady-state distribution, in contrast, is based on a thoroughly realistic mechanism of bursty transcription. The approach by Raj et al. (2006) and Grün et al. (2014) requires $r_{deact}/r_{deg} \to \infty$ and $r_{on}/r_{deact} < 1$. That means, the deactivation rate has to be substantially larger than the mRNA degradation rate and, simultaneously,

the transcription rate needs to be smaller than the gene deactivation rate. Here, we discuss the plausibility of these presumptions:
Schwanhäusser et al. (2011) showed that mRNA half-life is in median around $t_{1/2} = 9\,h$ (range: $1.61\,h$ to $40.47\,h$), which results in a degradation rate $r_{deg} = \log(2)/t_{1/2}$ of $0.077\,h^{-1} = 0.00128\,\text{min}^{-1}$ (range: $0.00718\,\text{min}^{-1}$ to $0.00029\,\text{min}^{-1}$). For $r_{deact}/r_{deg} \to \infty$, the mRNA degradation rate needs to become much smaller than the gene deactivation rate. Visual comparison shows that density curves of the PB and according NB distributions start to look similar for $r_{deact}/r_{deg} \approx 20,000$ (not shown here). Assuming a 20,000-fold larger gene deactivation rate results in $r_{deact} = 29.67\,\text{min}^{-1}$ (range: $143.51\,\text{min}^{-1}$ to $5.71\,\text{min}^{-1}$). This means that on average the gene switches approximately 30 times per minute into the off-state, i. e. on average the gene is in its active state for only two seconds. RNA polymerases proceed at 30 nt/sec (without pausing at approximately 70 nt/sec) (Darzacq et al., 2007). Genes have a length of hundreds to thousands of nucleotides. Thus, according to this speed and such length of genes, genes cannot be transcribed in such short phases. The switching model assumes the DNA to stay active during the whole transcription process of one (or more) mRNAs; as soon as the DNA turns inactive, all currently running transcriptions are stopped. In other words, although the NB distribution can mathematically be derived as a limiting steady-state distribution of the switching model, this entails biologically implausible assumptions.
This criticism is underpinned by the work of Suter et al. (2011) who derived ranges of the rates of the switching model experimentally and by calculations. Here, only so-called short-lived genes were taken into account. Thus, observed mRNA half-lives were on a smaller scale, mainly between 30 and 140 min, resulting in mRNA degradation rates between $0.005\,\text{min}^{-1}$ and $0.023\,\text{min}^{-1}$. At the same time, deactivation rates were found in the range between $0.1\,\text{min}^{-1}$ and $0.6\,\text{min}^{-1}$. Hence, their quotient is at maximum around 120 and thus nowhere close to infinity. Another mathematical assumption for deriving the NB limit distribution was that the transcription rate needed to be smaller than the deactivation rate. This is not confirmed by Suter et al. (2011) for most genes.

Biological plausibility of bursting model. Burst-like transcription has been discussed, e. g. Golding et al. (2005), Schwanhäusser et al. (2011) and Suter et al. (2011). We take a look at the inherent assumptions of the bursting model: The bursting rate r_{burst} represents the waiting time until the DNA turns open for transcription in addition to the time which the polymerase needs to transcribe. The model assumes that several polymerases attach simultaneously to the DNA and terminate transcription at the same time. By simplifying this part of the transcription process model, the problem of persisting DNA activation during the whole transcription process in the switching model is avoided.

Practical relevance. There is no unambiguous answer to the question of the most appropriate probability distribution for mRNA count data. Pragmatic reasons will often lead to NB distribution as already employed by many tools (see Table 4.1). However, the choice may depend on experimental techniques, the statistical analysis to be performed, and also differ between genes within the same dataset. For large read counts, even continuous distributions may be most suitable.

While statistics quantifies which model is the most plausible one from the data point of view, mathematical modeling points out which biological assumptions may implicitly be made when a particular distribution is used. Importantly, while the mechanistic model leads to a unique steady-state distribution, the reverse conclusion is not true. In general, the basic model and the corresponding Poisson distribution may appear too simple in most cases (both with respect to biological plausibility and the ability to describe measured sequencing data). The switching and bursting models are harder to distinguish. Apparently the data does not show the additional complexity that a PB distribution can in theory model better than a NB. Thus, from the mathematical point of view, in the cases considered their densities are of similar shape, such that the less complex NB model will often be preferred.Answering the question from the biological perspective may require measuring mRNA generation at a sufficiently small time resolution (e. g. Golding et al., 2005) to see whether several mRNA molecules are generated at once (bursting model) or in short successional intervals (switching model).

Additionally, we created two new mechanistic models for mRNA transcription and degradation and inferred possible steady state distributions. First we combined the basic and the bursting model to create a process called basic-bursting model that consists of a constant "background" transcription with occasional bursts leading to DEL distributed mRNA content. When trying to find a process that results into the PIG distribution as steady state distribution we created a similar dual process called IG bursting model where the rate of the basic "background" process is not constant but follows an IG distribution. These dual processes might be interesting to look at when investigating cells with two different alleles that transcribe mRNA with different mechanisms.

Taken together, we have identified mechanistic models for mRNA transcription and degradation with good interpretability, and established a link to mathematical representations by stochastic processes and steady-state count distributions. Specifically, the commonly used NB distribution model is supplied with a proper mechanistic model of the underlying biological process. The R package **scModels** overcomes a previous shortcoming in the implementation of the PB density. It provides a full toolbox for data simulation and parameter estimation, equipping users with the freedom to choose their models based on content-related, design-based or purely pragmatic motives.

5 Estimating Single-Cell Properties from Pooled Cell Data

Tissues are often heterogeneous in their single-cell molecular expression, and this can govern the regulation of cell fate. For the understanding of development and disease, it is important to quantify heterogeneity in a given tissue. We showed in the previous chapter the importance of selecting an appropriate distribution and knowing the inherent assumptions when selecting a tool for analysis. This is even more important if there is no suitable analysis method yet and a new one has to be developed or an existing one must be adapted to new conditions. In this chapter we aim to select a suitable distribution to adapt the stochastic profiling algorithm to discrete measurements. This algorithm has been developed to mathematically deconvolve joint measurements of several cells to their single-cell gene expression. Such joint measurements can be advantageous and their analysis might add to findings from single-cell or bulk analysis. Originally, it was developed by Bajikar et al. (2014) together with an early version of the R package **stochprofML**. Since then, we have developed and improved the implemented software by also extending the statistical model and the optimization procedure.

In this chapter we will present both the general idea of the statistical deconvolution method and highlight the extensions that we have added. Big parts of this chapter are based on and partly identical to the following preprint which is submitted and currently under revision:

> **Amrhein, L.** and Fuchs, C. (2020b). stochprofML: Stochastic Profiling Using Maximum Likelihood Estimation in R. *arXiv:2004.08809 [stat.AP]*.

With this we provided for the first time a complete description of the existing statistical model and the algorithm. During a collaboration with Stephan Tirier and Christian Conrad we included the possibility to handle pools of different sizes to the model. This is included in the following publication and as well is part of this chapter.

> Tirier, S. M., Park, J., Preußer, F., **Amrhein, L.**, Gu, Z., Steiger, S., Mallm, J.-P., Krieger, T., Waschow, M., Eismann, B., Gut, M., Gut, I. G., Rippe,

K., Schlesner, M., Theis, F., Fuchs, C., Ball, C. R., Glimm, H., Eils, R., and Conrad, C. (2019). Pheno-seq – linking visual features and gene expression in 3D cell culture systems. *Scientific Reports*, 9(12367).

The most important addition to the original model and the R package that we have added is the ability to model discrete counts by including the negative binomial distribution. In order to include the uncertainty on the true model parameters we included Bayesian inference. Some details in this chapter on these extensions are based on and partly identical to the following publication:

Amrhein, L. and Fuchs, C. (2020a). Stochastic Profiling of mRNA Counts Using HMC. *Proceedings of the 35th International Workshop on Statistical Modelling (IWSM).*

5.1 Background

Gene expression is stochastic. It can differ significantly between, e.g., types of cells or tissues, and between individuals. In that case, one refers to differential gene expression. In particular, cells can be differentially expressed between healthy and sick tissue samples from the same origin. Moreover, cells can differ even within a small tissue sample, e.g. within a tumor that consists of several mutated cell populations. More details are described in Section 2.1. Mathematically, two populations are regarded to be different if their mRNA numbers follow different probability distributions. If there is more than one population in a tissue, we call it heterogeneous. The expression of such tissues can be described by mixture models. Detecting and parametrizing heterogeneities is of utmost importance for understanding development and disease. The amount of mRNA molecules of a gene in a tissue sample can be assessed by various techniques such as microarray measurements (see Section 2.2.1) or sequencing (see Section 2.2.2). Bulk measurements are suitable for analyses like mean comparisons but make it difficult to describe in-bulk heterogeneity. To infer partial information about cell populations, bulk deconvolution methods like CIBERSORT (Newman et al., 2015) require the availability of so-called signature matrices. Measurements of single cells that were described in the models of the previous Chapter 4 yield the highest possible resolution. They are best suited for identification and description of heterogeneity in large and error-free datasets. In practice, however, single-cell data often comes along with high cost, effort and technical noise (Grün et al., 2014). Heterogeneity can still be revealed given sufficient sample size and additional information such as the expression of cell cycle genes (e. g. Buettner et al., 2015). In our work, we consider the case of comparatively small samples without further prior knowledge. Instead of considering single-cell data, we analyze the cumulative gene expression of small pools of randomly selected cells, see Section 2.2.5. The pool size should be large enough to substantially reduce measurement error and cost, and at the same time small enough such that heterogeneity is still identifiable. The analysis of such small cell pools could add additional information that is lost in single cell measurements due to the stress in which the cells find themselves once they are separated from their tissue.

Such new kind of data requires new analysis tools. We thus developed the stochastic profiling algorithm to infer single-cell regulatory states from small pools of cells (Bajikar et al., 2014). In contrast to previously existing deconvolution methods, which were not tailored to small cell pools, we neither require a priori knowledge about the mixing weights (such as Erkkilä et al., 2010, Shen-Orr et al., 2010) nor about expression profiles (such as Abbas et al., 2009, Gong et al., 2011). Only Wang et al. (2016) perform unsupervised deconvolution for clusters of genes, however with the aim to find marker genes. Several of these methods are implemented in the R package `CellMix` (Gaujoux and Seoighe, 2013), but for the above reasons, they are not directly comparable. In Bajikar et al. (2014), it is demonstrated on synthetic data how stochastic profiling led to more accurate estimates than competing approaches. Recently many tools were developed with the aim to deconvolute bulk measurements using the available huge datasets of single-cell data or purified bulk samples (such as Aliee and Theis, 2020, Frishberg et al., 2019, Hunt et al., 2018). However, deconvolution without any basis such as purified expression datasets of sub populations or other prior knowledge is much harder. Here we present the stochastic profiling algorithm that blindly deconvolves the joint measurements purely by applying a combinatorial mixture model. In Bajikar et al. (2014), **stochprofML** is applied to measurements from human breast epithelial cells and revealed the functional relevance of the heterogeneous expression of a particular gene. Fluorescence in situ hybridization confirmed that the computationally identified population fractions corresponded to experimentally detected transcriptional populations. In another study (Tirier et al., 2019), we applied the algorithm to clonal tumor spheroids of colorectal cancer. There, a single tumor cell was cultured, and after several rounds of replication, each resulting spheroid was imaged and sequenced. However, pool sizes differed between tissue samples as each spheroid contained a different number of cells ranging from less than ten to nearly 200 cells. Therefore, we extended **stochprofML** to be able to handle pools of different sizes.

Since recent technological advances make small-pool sequencing possible, resulting in discrete small-pool mRNA counts (see 2.2.5 and Singh et al., 2019), we develop the stochastic profiling algorithm further to apply it to novel discrete count data. Certainly, it is possible to simply keep on using the continuous model, but you have to be aware that you will make wrong model assumptions regarding the data and therefore you cannot trust the results as much as with a model without violations of the underlying assumptions.

In this chapter, we present such modeling extensions alongside numerical and computational details. We include a complete description of the existing continuous algorithm since the new discrete version is based on it and various extensions are also applied to the continuous version. We explore the performance of the algorithm in simulation studies for various settings, especially in the realistic case of uncertainty about the pool size. To expand the range of applications, we propose a test for significant differences between the estimated populations and inference of original pool compositions. We include a discrete model using NB distributions and compare the **stochprofML** parameter inference with Bayesian parameter inference using the programming language Stan. An application of the discrete version to real-world data is included in Chapter 6.

5.2 Statistical Model Underlying Stochastic Profiling

In this section, we present the statistical convolution model and derive the likelihood functions of the parameters including recent extensions. Note that we will use a combinatorial mixture since we aim for a blind convolution model that does not need any prior input information on the contained subpopulations or their fractions. After a first description of the nomenclature, we introduce basic statistical descriptions of univariate single-cell gene expression. The complexity of the model is increased step by step: First, we account for cell-to-cell heterogeneity through the use of mixture distributions. Then, we extend the modeling from single-cell to small-pool measurements by introducing convolutions of statistical distributions. Finally, we calculate the likelihood that is needed for parameter inference.

5.2.1 Notation

Suppose there are k (tissue) samples, indexed by $i \in \{1, \ldots, k\}$. From each tissue sample i, we collect a pool of a known number of cells. The cells are either indexed by $j \in \{1, \ldots, n\}$ if the cell pool size is the same in all measurements, or, as possible in the latest implementation, by $j_i \in \{1, \ldots, n_i\}$ in case cell pool sizes vary between measurements. In the latter, more general case, the cell numbers are variable over the k cell pools and summarized by the vector $\vec{n} = (n_1, \ldots, n_k)$. From each sample, the gene expression of m genes is measured, indexed by $g \in \{1, \ldots, m\}$. We assume that each cell stems from one out of T cell populations, indexed by $h \in \{1, \ldots, T\}$. If $T > 1$ in the set of all cells of interest, the tissue is called heterogeneous. The notation is illustrated in Figure 5.1. Biologically, the different cell populations correspond to different regulatory states or — especially in the context of cancer — to different (sub-)clones. For example, there might be two populations within a considered tissue: one occupying a basal regulatory state, where the expression of genes is at a low level, and one from a second regulatory state, where genes are expressed at a higher level.

5.2.2 Single-Cell Models of Heterogeneous Gene Expression

As described in Section 2.2, there are various technologies to measure gene expression. Microarrays (as considered in previous applications of stochastic profiling, see Janes et al. (2010) and Bajikar et al. (2014)) measure relative gene expression, which is appropriately described in terms of continuous probability distributions. Sequencing experiments produce discrete molecule counts. However, if these numbers are large, or if preprocessing blurs the discrete character of the data, one often still describes such sequencing output by continuous probability distributions as well. Conditioned

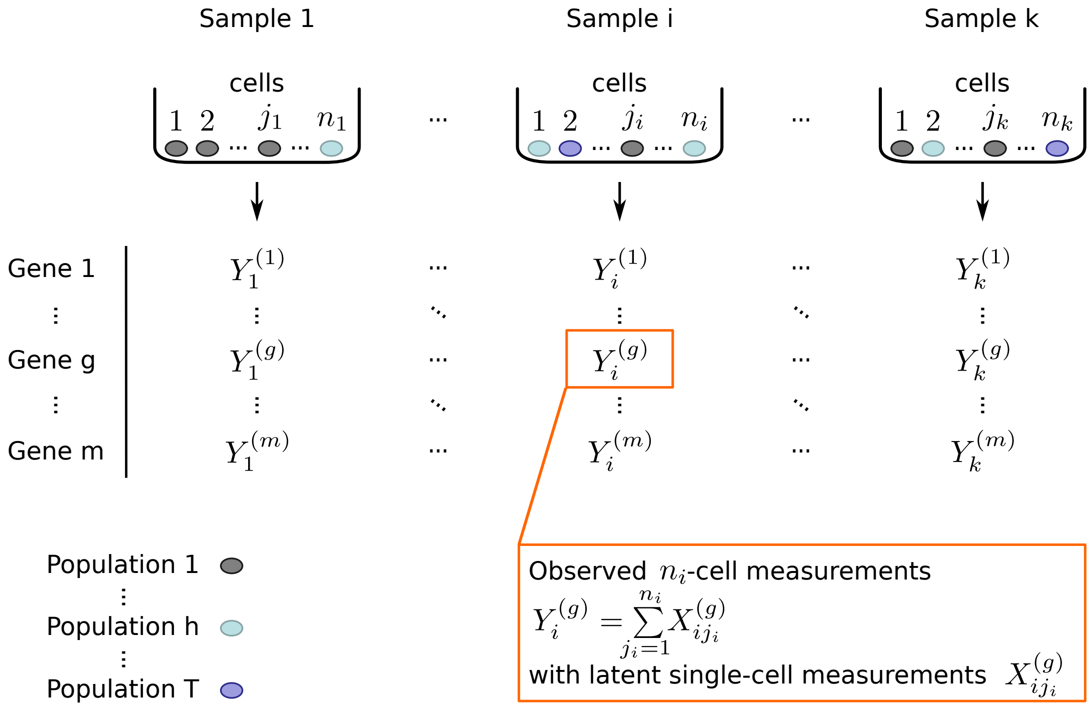


Figure 5.1: Experimental design of pooling cells into samples, measuring the pooled gene expression across several genes for which identical population structures are assumed. The table illustrates the index notation of (tissue) samples, single cells, populations and genes as well as observed and latent measurements.

on the cell population, originally two continuous choices for the single-cell distribution of the expression of one gene, the lognormal (see Definition A.1) and the exponential distribution (see Definition A.4). In general, the lognormal distribution is an appropriate description of continuous gene expression (Bengtsson, 2005). With its two parameters, it is more flexible than the exponential distribution. However, the lognormal distribution cannot model zero gene expression. In case of zeros in the data, it could be modified by adding very small values such as 0.0001, or one uses the exponential distribution to model this kind of expression.

Recently, we added the negative binomial (NB) distribution (see Definition A.8) as a discrete single-cell distribution. The NB distribution with its two parameters is flexible and can model zero expression at the same time. In the previous chapters, especially in Chapter 4 we have already discussed in detail the NB distribution as a distribution for discrete single-cell sequencing measurements.

Given T cell populations, the expression of one gene is described by a stochastic mixture model described by a T-fold mixture distribution (see Definition 3.1). Let $(p_1, \ldots, p_T)$ with $p_1 + \ldots + p_T = 1$ denote the fractions of populations in the overall set of cells. Different combinations of lognormal and the exponential distributions lead to the following three continuous mixture models, offered by **stochprofML**:

Lognormal-lognormal (LN-LN): Each population h is represented by a lognormal distribution with population-specific parameter μ_h (different for each population h) and identical σ for all T populations. The single-cell expression X that originates from such a mixture of populations then follows

$$X \sim \begin{cases} \mathrm{LN}(\mu_1, \sigma^2) & \text{with probability } p_1 \\ \vdots & \\ \mathrm{LN}(\mu_h, \sigma^2) & \text{with probability } p_h \\ \vdots & \\ \mathrm{LN}(\mu_T, \sigma^2) & \text{with probability } \left(1 - \sum_{h=1}^{T-1} p_h\right). \end{cases}$$

Relaxed lognormal-lognormal (rLN-LN): This model is similar to the LN-LN model, but each population h is represented by a lognormal distribution with a different parameter set (μ_h, σ_h). The single-cell expression X follows

$$X \sim \begin{cases} \mathrm{LN}(\mu_1, \sigma_1^2) & \text{with probability } p_1 \\ \vdots & \\ \mathrm{LN}(\mu_h, \sigma_h^2) & \text{with probability } p_h \\ \vdots & \\ \mathrm{LN}(\mu_T, \sigma_T^2) & \text{with probability } \left(1 - \sum_{h=1}^{T-1} p_h\right). \end{cases}$$

Exponential-lognormal (EXP-LN): Here, one population is represented by an exponential distribution with parameter λ, and all remaining $T-1$ populations are modeled by lognormal distributions analogously to LN-LN, i.e. with population-specific parameters μ_h and identical σ. The single-cell expression X then follows

$$X \sim \begin{cases} \mathrm{LN}(\mu_1, \sigma^2) & \text{with probability } p_1 \\ \vdots & \\ \mathrm{LN}(\mu_h, \sigma^2) & \text{with probability } p_h \\ \vdots & \\ \mathrm{LN}(\mu_{T-1}, \sigma^2) & \text{with probability } p_{T-1} \\ \mathrm{EXP}(\lambda) & \text{with probability } \left(1 - \sum_{h=1}^{T-1} p_h\right). \end{cases}$$

The LN-LN model is a special case of the rLN-LN model. It assumes identical σ across all populations. Biologically, this assumption is motivated by the fact that, for the lognormal distribution, identical σ lead to identical coefficient of variation

$$\mathrm{CV}(X) = \frac{\sqrt{\mathrm{Var}(X)}}{\mathrm{E}(X)} = \sqrt{\exp(\sigma^2) - 1}$$

even for different values of μ. In other words, the linear relationship between the mean expression and the standard deviation is maintained across cell populations in the LN-LN model. The appropriateness of the different mixture models can be discussed both biologically and in terms of statistical model choice.

In the case of discrete gene expression we included the following NB mixture model, which is now also implemented in the **stochprofML** package:

Negative binomial-negative binomial (NB-NB): Each population h is represented by a NB distribution with population-specific parameter set $(\mu_h,\, r_h)$. Note that here, we use the alternative parametrization of the NB distribution via the mean μ as described in Definition A.8 to be able to order the populations directly by their mean expression analogously to the continuous distributions. The single-cell expression X that originates from such a mixture of populations follows

$$X \sim \begin{cases} \mathrm{NB}(\mu_1, r_1) & \text{with probability } p_1 \\ \vdots & \\ \mathrm{NB}(\mu_h, r_h) & \text{with probability } p_h \\ \vdots & \\ \mathrm{NB}(\mu_T, r_T) & \text{with probability } \left(1 - \sum_{h=1}^{T-1} p_h\right). \end{cases}$$

Within one set of genes under consideration, we assume that the same type of model (LN-LN, rLN-LN, EXP-LN, NB-NB) is appropriate for all genes. The parameter values, however, may differ. With Definition 3.1 the the single-cell gene expression $X^{(g)}$ for gene g by a T-fold mixture distribution with PDF/PMF in the continuous/discrete case is given by

$$f_{\text{T-pop}}\left(x^{(g)} |\ \boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right) =$$
$$p_1 f_1\left(x^{(g)}|\theta_1^{(g)}\right) + \ldots + p_h f_h\left(x^{(g)}|\theta_h^{(g)}\right) + \ldots + \left(1 - \sum_{h=1}^{T-1} p_h\right) f_T\left(x^{(g)}|\theta_T^{(g)}\right),$$

where f_h with $h \in \{1, \ldots, T\}$ represents the PDF/PMF of population h that here are assumed to be in the continuous case either lognormal or exponential or in the discrete case NB. $\boldsymbol{\theta}^{(g)} = \{\theta_1^{(g)}, \ldots, \theta_T^{(g)}\}$ are the distribution parameters of the T populations for gene g.

Example 5.1 (Mixture of two populations - Part 1) *We exemplify the two-population case. Here, the PDF/PMF of the mixture distribution for gene g reads*

$$f_{\textit{2-pop}}(x^{(g)}|\boldsymbol{\theta}^{(g)}) = p f_1(x^{(g)}|\theta_1^{(g)}) + (1-p) f_2(x^{(g)}|\theta_2^{(g)}),$$

where p is the probability of the first population. The univariate distributions $f_1^{(g)}$ *and* $f_2^{(g)}$ *depend on the chosen model :*
LN-LN: $f_1(x^{(g)}|\theta_1^{(g)}) = f_{LN}(x^{(g)}|\mu_1^{(g)}, \sigma^2)$ *and* $f_2(x^{(g)}|\theta_2^{(g)}) = f_{LN}(x^{(g)}|\mu_2^{(g)}, \sigma^2)$*, i.e. there are four unknown parameters:* $p, \mu_1^{(g)}, \mu_2^{(g)}$ *and* σ^2*.*
rLN-LN: $f_1(x^{(g)}|\theta_1^{(g)}) = f_{LN}(x^{(g)}|\mu_1^{(g)}, {\sigma_1}^2)$ *and* $f_2(x^{(g)}|\theta_2^{(g)}) = f_{LN}(x^{(g)}|\mu_2^{(g)}, {\sigma_2}^2)$ *i.e.*

there are five unknown parameters: $p, \mu_1^{(g)}, \mu_2^{(g)}, {\sigma_1}^2$ *and* ${\sigma_2}^2$*.*
EXP-LN: $f_1(x^{(g)}|\theta_1^{(g)}) = f_{LN}(x^{(g)}|\mu^{(g)}, \sigma^2)$ *and* $f_2(x^{(g)}|\theta_2^{(g)}) = f_{EXP}(x^{(g)}|\lambda^{(g)})$*. i.e. there are four unknown parameters:* $p, \mu^{(g)}$*,* σ^2 *and* $\lambda^{(g)}$*.*
Note that although each lognormal population has its individual σ*, these* σ*-values remain identical across genes in all models.*
NB-NB: $f_1(x^{(g)}|\theta_1^{(g)}) = f_{\mathrm{NB}}(x^{(g)}|\mu_1^{(g)}, r_1^{(g)})$ *and* $f_2(x^{(g)}|\theta_2^{(g)}) = f_{\mathrm{NB}}(x^{(g)}|\mu_2^{(g)}, r_2^{(g)})$ *i.e. there are five unknown parameters:* $p, \mu_1^{(g)}, \mu_2^{(g)}, {r_1}^{(g)}$ *and* ${r_2}^{(g)}$*.*

5.2.3 Small-Pool Models of Heterogeneous Gene Expression

Stochastic profiling is tailored to analyze gene expression measurements of small pools of cells, beyond the analysis of standard single-cell gene expression data. In other words, the single-cell gene expression $X_{ij_i}^{(g)}$ described above is assumed latent. Instead, consider observations

$$Y_i^{(g)} = \sum_{j_i=1}^{n_i} X_{ij_i}^{(g)} \tag{5.1}$$

for $i = 1, \ldots, k$, which represent the overall gene expression of the ith cell pool for gene g. In the first version of **stochprofML**, pools had to be of equal size n, i.e. for each measurement $Y_i^{(g)}$ one had to extract the same number of cells from each tissue sample. This was a restrictive assumption from the experimental point of view. One recent extension of **stochprofML** allows each cell pool i to contain a different number n_i of cells (see also Figures 5.1 and 5.2). This was done during a collaboration with Stephan Tirier and Christian Conrad (Tirier et al., 2019) where we applied the algorithm to clonal tumor spheroids of colorectal cancer. A single tumor cell was cultured, and after several rounds of replication, each resulting spheroid was imaged and sequenced. However, pool sizes differed between tissue samples as each spheroid contained a different number of cells ranging from less than ten to nearly 200 cells. The algorithm aims to estimate the single-cell population parameters despite the fact that measurements are available only in convoluted form. To that end, the likelihood function of the parameters in the convolution model (5.1) is derived, where the gene expression of the single cells is assumed to be independent within a tissue sample. For better readability, we suppress for now the superscript (g) and introduce it again later.

Next, we derive the distribution of Y_i, the PDF of n-cell measurements of T cell populations. We derive this convolution of mixed distributions in four steps: We start with the simplest case of 2-cell measurements in the presence of two populations. Then, we continue with 2-cell samples and three populations. Next, the cell number is increased to n and finally the population number is raised to T. Here, we only look at the continuous case. However, to calculate the resulting PMF in the discrete case similar: all integrals are replaced by sums.

PDF of 2-Cell Measurements of Two Populations ($n = 2$, $T = 2$) First, the PDF of a measurement y of a 2-cell pool is derived, i.e. of $Y = X_1 + X_2$. Assume

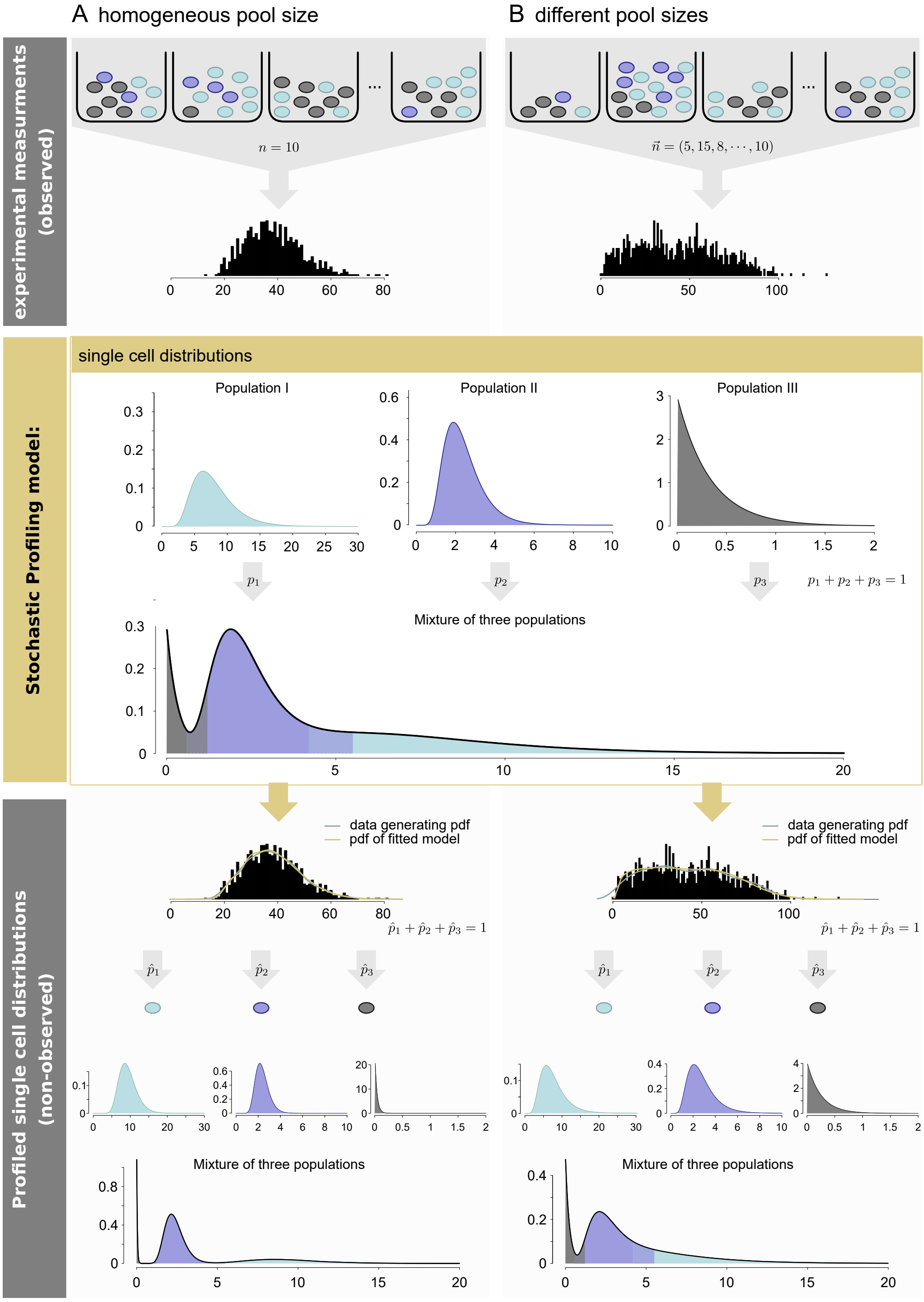

A homogeneous pool size
B different pool sizes
experimental measurements (observed)
$n = 10$
$\vec{n} = (5, 15, 8, \cdots, 10)$
0 20 40 60 80
0 50 100
Stochastic Profiling model:
single cell distributions
Population I
Population II
Population III
p_1
p_2
p_3
$p_1 + p_2 + p_3 = 1$
Mixture of three populations
Profiled single cell distributions (non-observed)
data generating pdf
pdf of fitted model
$\hat{p}_1 + \hat{p}_2 + \hat{p}_3 = 1$
$\hat{p}_1$
$\hat{p}_2$
$\hat{p}_3$
Mixture of three populations

Figure 5.2: Stochastic Profiling can be performed either on measurements of (A) homogeneous pool size of n cells or of (B) different pool sizes given by the cell number vector $\vec{n}$. In both cases, the **stochprofML** algorithm estimates the parameters for the specified number of populations from pooled data, leading to inferred single-cell distributions for each population. Finally the mixture distribution can be summarized. Section 5.2.3 contains a description how this density is visualized in the case of measurements of different pool sizes.

we know that two cell populations are present in the tissue, and each of them is described by an individual distribution. In this section,the univariate population distributions is denoted by $\mathcal{D}_h$, $h = 1, \ldots, T = 2$ and can in general be replaced by any distribution. For now, we consider for $j = 1, 2$

$$X_j \overset{iid}{\sim} \begin{cases} \mathcal{D}_1 & \text{with probability } p_1 \\ \mathcal{D}_2 & \text{with probability } 1 - p_1, \end{cases}$$

where $p_1 \in [0, 1]$. Hence, the , PDF of each X_j is a mixture distribution (see Definition 3.1)

$$f_X(x) = p_1 f_{\mathcal{D}_1}(x) + p_2 f_{\mathcal{D}_2}(x)$$

with $p_2 = 1 - p_1$. To determine the distribution of Y, we use the convolution (see Definition 3.2) of the single-cell PDFs, which are the same functions f_X for both X_1 and X_2:

$$\begin{aligned} f_Y(y) &= \int_0^y f_X(x_1) f_X(y - x_1) dx_1 \\ &= \int_0^y \left(\Big[p_1 f_{\mathcal{D}_1}(x_1) + p_2 f_{\mathcal{D}_2}(x_1) \Big] \Big[p_1 f_{\mathcal{D}_1}(y - x_1) + p_2 f_{\mathcal{D}_2}(y - x_1) \Big] \right) dx_1 \\ &= \int_0^y \Bigg(p_1^2 f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_1}(y - x_1) + p_2^2 f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_2}(y - x_1) \\ &\qquad + p_1 p_2 f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_2}(y - x_1) + p_2 p_1 f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_1}(y - x_1) \Bigg) dx_1 \\ &= p_1^2 \int_0^y f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_1}(y - x_1) dx_1 + p_2^2 \int_0^y f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_2}(y - x_1) dx_1 \\ &\qquad + p_1 p_2 \int_0^y f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_2}(y - x_1) dx_1 + p_2 p_1 \int_0^y f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_1}(y - x_1) dx_1. \end{aligned}$$

Each of these integrals $\int_0^y f_{\mathcal{D}_i}(x_1) f_{\mathcal{D}_j}(y - x_1) dx_1$ is the PDF of a random variable $Z_1 + Z_2$ evaluated at y, where $Z_1 \sim \mathcal{D}_i$ and $Z_2 \sim \mathcal{D}_j$ are independent. This holds for both $i \neq j$ and $i = j$. Denoting this density by $f_{i,j}$, we get

$$f_Y(y) = \sum_{i=1}^{2} \sum_{j=1}^{2} p_i p_j f_{i,j}(y).$$

An alternative formulation is

$$f_Y(y) = \sum_{\ell_1=0}^{2} \binom{2}{\ell_1} p_1^{\ell_1} p_2^{\ell_2} f_{(\ell_1,\ell_2)}(y), \tag{5.2}$$

where ℓ_1 and $\ell_2 = 2 - \ell_1$ show how often a cell of population 1 and 2 is present in the pool. The two PDFs $f_{(\ell_1,\ell_2)}$ and $f_{i,j}$ are directly connected: $f_{(\ell_1,\ell_2)}$ considers *how often* populations 1 and 2 are represented, and $f_{i,j}$ denotes *which* populations are present. For example, $f_{(1,1)}(y) = f_{1,2}(y)$ and $f_{(0,2)}(y) = f_{2,2}(y)$.

PDF of 2-Cell Measurements of Three Populations ($n = 2$, $T = 3$) Next, we derive the PDF of a measurement y of a 2-cell pool, i.e. of $Y = X_1 + X_2$. Now, we assume three cell populations to be present in the tissue. Again, each of them is described by an individual distribution $\mathcal{D}_h$ for $h = 1, \ldots, T = 3$:

$$X_j \overset{iid}{\sim} \begin{cases} \mathcal{D}_1 & \text{w.p. } p_1 \\ \mathcal{D}_2 & \text{w.p. } p_2 \\ \mathcal{D}_3 & \text{w.p. } 1 - p_1 - p_2, \end{cases}$$

for $j = 1, 2$ where $p_1, p_2 \in [0, 1]$ and $p_1 + p_2 \leq 1$. Hence, the mixture PDF of each X_j is

$$f_X(x) = p_1 f_{\mathcal{D}_1}(x) + p_2 f_{\mathcal{D}_2}(x) + p_3 f_{\mathcal{D}_3}(x)$$

with $p_3 = 1 - p_1 - p_2$. To determine the distribution of $Y = X_1 + X_2$, we again use the convolution of the single-cell PDFs:

$$\begin{aligned} f_Y(y) &= \int_0^y f_X(x_1) f_X(y - x_1) dx_1 \\ &= \int_0^y \Bigg(\Big[p_1 f_{\mathcal{D}_1}(x_1) + p_2 f_{\mathcal{D}_2}(x_1) + p_3 f_{\mathcal{D}_3}(x_1) \Big] \\ &\qquad \times \Big[p_1 f_{\mathcal{D}_1}(y - x_1) + p_2 f_{\mathcal{D}_2}(y - x_1) + p_3 f_{\mathcal{D}_3}(y - x_1) \Big] \Bigg) dx_1 \\ &= \int_0^y \Bigg(p_1^2 f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_1}(y - x_1) \\ &\qquad + p_2^2 f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_2}(y - x_1) + p_3^2 f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_3}(y - x_1) \\ &\qquad + p_1 p_2 f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_2}(y - x_1) + p_2 p_1 f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_1}(y - x_1) \\ &\qquad + p_1 p_3 f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_3}(y - x_1) + p_3 p_1 f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_1}(y - x_1) \\ &\qquad + p_2 p_3 f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_3}(y - x_1) + p_3 p_2 f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_2}(y - x_1) \Bigg) dx_1, \end{aligned}$$

leading to

$$\begin{aligned} f_Y(y) = p_1^2 &\int_0^y f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_1}(y - x_1) dx_1 \\ &+ p_2^2 \int_0^y f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_2}(y - x_1) dx_1 + p_3^2 \int_0^y f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_3}(y - x_1) dx_1 \\ &+ p_1 p_2 \int_0^y f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_2}(y - x_1) dx_1 + p_2 p_1 \int_0^y f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_1}(y - x_1) dx_1 \\ &+ p_1 p_3 \int_0^y f_{\mathcal{D}_1}(x_1) f_{\mathcal{D}_3}(y - x_1) dx_1 + p_3 p_1 \int_0^y f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_1}(y - x_1) dx_1 \\ &+ p_2 p_3 \int_0^y f_{\mathcal{D}_2}(x_1) f_{\mathcal{D}_3}(y - x_1) dx_1 + p_3 p_2 \int_0^y f_{\mathcal{D}_3}(x_1) f_{\mathcal{D}_2}(y - x_1)) dx_1. \end{aligned}$$

Once more, we make use of the fact that $\int_0^y f_{\mathcal{D}_i}(x_1) f_{\mathcal{D}_j}(y-x_1)dx_1$ is the PDF of the sum Z_1+Z_2 of two independent random variables, where $Z_1 \sim \mathcal{D}_i$ and $Z_2 \sim \mathcal{D}_j$ (now with $i,j \in \{1,2,3\}$). As before, we denote this density by $f_{i,j}$. Overall, we obtain

$$f_Y(y) = \sum_{i=1}^{3}\sum_{j=1}^{3} p_i p_j f_{i,j}(y),$$

or alternatively

$$f_Y(y) = \sum_{\ell_1=0}^{2}\sum_{\ell_2=0}^{2-\ell_1} \binom{2}{\ell_1}\binom{2-\ell_1}{\ell_2} p_1^{\ell_1} p_2^{\ell_2} p_3^{\ell_3} f_{(\ell_1,\ell_2,\ell_3)}(y), \tag{5.3}$$

where $\ell_1, \ell_2, \ell_3 = 2-\ell_1-\ell_2$ show how often cells of population 1, 2 and 3 are present in the pool. Again, $f_{(\ell_1,\ell_2,2-\ell_1-\ell_2)}(y)$ is connected to $f_{i,j}$. For example, $f_{(0,1,1)}(y) = f_{2,3}(y)$ and $f_{(2,0,0)}(y) = f_{1,1}(y)$.

PDF of n-Cell Measurements of Three Populations (n arbitrary, $T = 3$) Next, we suppose that we measure pools of n cells originating from three cell populations. Let $Y = X_1 + \ldots + X_n$. Then Equation (5.3) turns into

$$f_Y(y) = \sum_{\ell_1=0}^{n}\sum_{\ell_2=0}^{n-\ell_1} \binom{n}{\ell_1}\binom{n-\ell_1}{\ell_2} p_1^{\ell_1} p_2^{\ell_2} p_3^{\ell_3} f_{(\ell_1,\ell_2,\ell_3)}(y), \tag{5.4}$$

where $p_3 = 1 - p_1 - p_2$ and $\ell_3 = n - \ell_1 - \ell_2$.

PDF of n-Cell Measurements of T Populations (n and T arbitrary) Finally, we extend Equation (5.4) to the most general case, where n-cell pools are measured from a tissue that consists of T cell populations. Here, we obtain

$$f_Y(y) = \sum_{\ell_1=0}^{n}\sum_{\ell_2=0}^{n-\ell_1}\cdots\sum_{\ell_{T-1}=0}^{n-\ell_1-\ldots-\ell_{T-1}} \binom{n}{\ell_1}\binom{n-\ell_1}{\ell_2}\cdots\binom{n-\ell_1-\ldots-\ell_{T-2}}{\ell_{T-1}} p_1^{\ell_1}\ldots p_T^{\ell_T} f_{(\ell_1,\ldots,\ell_T)}(y),$$

where $p_T = 1 - p_1 - \ldots - p_{T-1}$ and $\ell_T = n - \ell_1 - \ldots - \ell_{T-1}$.
Since $f_{(\ell_1,\ldots,\ell_T)}(y)$ in general can be describe a PDF but also a PMF, the following is valid for both – continuous and discrete – models. Using Identity 5 leads to the final PDF/PMF $f_{n_i}(y_i|\boldsymbol{\theta},\boldsymbol{p})$ of an observation y_i which represents the overall gene expression from sample i (consisting of n_i cells)

$$f_{n_i}\left(y_i|\ \boldsymbol{\theta},\boldsymbol{p}\right) = \sum_{\ell_1=0}^{n_i}\sum_{\ell_2=0}^{n_i-\ell_1}\cdots\sum_{\ell_{T-1}=0}^{n_i-\sum_{h=1}^{T-2}\ell_h} \binom{n_i}{\ell_1,\ell_2,\ldots,\ell_T} p_1^{\ell_1}p_2^{\ell_2}\cdots p_T^{\ell_T} f_{(\ell_1,\ell_2,\ldots,\ell_T)}\left(y_i|\boldsymbol{\theta}\right), \tag{5.5}$$

where $\ell_T = n_i - \sum_{h=1}^{T-1} \ell_h$ and $p_T = 1 - \sum_{h=1}^{T-1} p_h$.
The terms $\binom{n_i}{\ell_1,\ldots,\ell_T} p_1^{\ell_1} \cdots p_T^{\ell_T}$ are probabilities arising from the multinomial distribution and therefore can be seen as multinomial weights of the densities $f_{(\ell_1,\ldots,\ell_T)}(y)$.Note that, here, $f_{(\ell_1,\ell_2,\ldots,\ell_T)}$ describes the PDF/PMF of a pool of n_i cells with *known* composition of the single populations, i.e. it is known that there are ℓ_1 cells from population 1, ℓ_2 cells from population 2 etc. Therefore, $\binom{n_i}{\ell_1,\ell_2,\ldots,\ell_T} p_1^{\ell_1} p_2^{\ell_2} \cdots p_T^{\ell_T}$ represents the multinomial probability of obtaining exactly this composition $(\ell_1,\ldots,\ell_T)$ using the multinomial coefficient $\binom{n_i}{\ell_1,\ell_2,\ldots,\ell_T} = n_i!/(\ell_1!\ldots\ell_T!)$. Equation (5.5) sums up over all possible compositions $(\ell_1,\ldots,\ell_T)$ with $\ell_1,\ldots,\ell_T \in \mathbb{N}_0$ and $\ell_1 + \ldots + \ell_T = n_i$. Taken together, $f_{n_i}(y_i|\boldsymbol{\theta},\boldsymbol{p})$ determines the PDF/PMF of y_i with respect to each possible combination of n_i cells of T populations.
Thus, the calculation of $f_{n_i}(y_i|\boldsymbol{\theta},\boldsymbol{p})$ requires knowledge of $f_{(\ell_1,\ell_2,\ldots,\ell_T)}(y_i|\boldsymbol{\theta})$. The derivation of this PDF/PMF depends on the choice of the single-cell model (LN-LN, rLN-LN, EXP-LN or NB-NB)that was made for X_{ij_i}.

LN-LN:

$$f_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\boldsymbol{\theta}) = f^{\text{LN-LN}}_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\mu_1,\ldots,\mu_h,\ldots,\mu_T,\sigma^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{in_i}$ of n_i independent random variables with

$$X_{ij_i} \sim \begin{cases} \text{LN}(\mu_1,\sigma^2) & \text{if } 1 \leq j_i \leq J_1 \\ \vdots & \\ \text{LN}(\mu_h,\sigma^2) & \text{if } J_{h-1} < j_i \leq J_h \\ \vdots & \\ \text{LN}(\mu_T,\sigma^2) & \text{if } J_{T-1} < j_i \leq J_T = n_i, \end{cases}$$

with $J_1 = \ell_1,\ldots,J_h = \ell_1 + \ell_2 + \ldots + \ell_h,\ldots,J_T = \ell_1 + \ell_2 + \ldots + \ell_T = n_i$. Y_i is the convolution of random variables $X_{i1},\ldots,X_{in_i}$, which is here the convolution of T sub-convolutions: a convolution of ℓ_1 times $\text{LN}(\mu_1,\sigma^2)$, plus a convolution of ℓ_2 times $\text{LN}(\mu_2,\sigma^2)$, and so on, up to a convolution of ℓ_T times $\text{LN}(\mu_T,\sigma^2)$.
Hence there is no analytically explicit form for the convolution of lognormal random variables it is approximated using the method by Fenton (1960) as described in Example 3.11. In the **stochprofML** package, this approximation is implemented in the function `d.sum.of.lognormals()`. The overall PDF given in (5.5) of the LN-LN model, $f^{\text{LN-LN}}_{n_i}(y_i|\mu_1,\ldots,\mu_h,\ldots,\mu_T,\sigma^2,p_1,\ldots,p_h,\ldots,p_T)$, with $p_1 + \cdots + p_T = 1$, is computed through `d.sum.of.mixtures.LNLN()`.

rLN-LN:

$$f_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\boldsymbol{\theta}) = f^{\text{rLN-LN}}_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\mu_1,\ldots,\mu_h,\ldots,\mu_T,\sigma_1^2,\ldots,\sigma_h^2,\ldots,\sigma_T^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{in_i}$ of n_i independent random variables with

$$X_{ij_i} \sim \begin{cases} \mathrm{LN}(\mu_1, \sigma_1^2) & \text{if } 1 \leq j_i \leq J_1 \\ \vdots & \\ \mathrm{LN}(\mu_h, \sigma_h^2) & \text{if } J_{h-1} < j_i \leq J_h \\ \vdots & \\ \mathrm{LN}(\mu_T, \sigma_T^2) & \text{if } J_{T-1} < j_i \leq J_T = n_i, \end{cases}$$

with $J_1 = \ell_1, \ldots, J_h = \ell_1 + \ell_2 + \ldots + \ell_h, \ldots, J_T = \ell_1 + \ldots + \ell_T = n_i$. Again, $f^{\text{rLN-LN}}_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}$ is approximated using the method by Fenton (1960), using the function that was used in the LN-LN model. The overall PDF given in (5.5) of the rLN-LN model, $f^{\text{rLN-LN}}_{n_i}(y_i|\mu_1, \ldots, \mu_h, \ldots, \mu_T, \sigma_1^2, \ldots, \sigma_h^2, \ldots, \sigma_T^2, p_1, \ldots, p_h, \ldots, p_T)$, with $p_1 + \cdots + p_T = 1$, is implemented in the **stochprofML** package via the function `d.sum.of.mixtures.rLNLN()`.

EXP-LN:

$$f_{(\ell_1,\ell_2,\ldots,\ell_T)}(y_i|\boldsymbol{\theta}) = f^{\text{EXP-LN}}_{(\ell_1,\ell_2,\ldots,\ell_T)}(y_i|\lambda, \mu_1, \ldots, \mu_{T-1}, \sigma^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{in_i}$ of n_i independent random variables with

$$X_{ij_i} \sim \begin{cases} \mathrm{LN}(\mu_1, \sigma^2) & \text{if } 1 \leq j_i \leq J_1 \\ \vdots & \\ \mathrm{LN}(\mu_h, \sigma^2) & \text{if } J_{h-1} < j_i \leq J_h \\ \vdots & \\ \mathrm{LN}(\mu_{T-1}, \sigma^2) & \text{if } J_{T-2} < j_i \leq J_{T-1} \\ \mathrm{EXP}(\lambda) & \text{if } J_{T-1} < j_i \leq J_T = n_i, \end{cases}$$

with $J_1 = \ell_1, \ldots, J_h = \ell_1 + \ell_2 + \ldots + \ell_h, \ldots, J_T = \ell_1 + \ldots + \ell_T = n_i$. On the highest level, this is a convolution consisting of two sub-convolutions. The first sub-convolution, convolves ℓ_t times EXP(λ). As shown in Example 3.9, the sum of independent exponentially distributed random variables with equal intensity parameter follows an Erlang distribution which is a special case of the gamma distribution (see Definition A.3). The second sub-convolution consists of the convolution of all remaining lognormals. For this we reuse the convoluted lognormals from above where the method by Fenton (1960) is used to approximate this convolution by another lognormal distribution. Taken together, the PDF for the EXP-LN mixture model is approximated by the convolution of one Erlang (or gamma) distribution and one lognormal distribution. The PDF for this convolution is not known in analytically explicit form but expressed in terms of an integral that is solved numerically through the function `lognormal.exp.convolution()`. Its computation thus takes substantially longer in terms of run time than for LN-LN. In the **stochprofML** package the function `d.sum.of.mixtures.EXPLN()` contains the implementation of the overall PDF given in (5.5) of the EXP-LN model, $f^{\text{EXP-LN}}_{n_i}(y_i|\lambda, \mu_1, \ldots, \mu_h, \ldots, \mu_T, \sigma^2, p_1, \ldots, p_h, \ldots, p_T)$, with $p_1 + \cdots + p_T = 1$.

NB-NB:

$$f_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\boldsymbol{\theta}) = f^{\text{NB-NB}}_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}(y_i|\mu_1,\ldots,\mu_h,\ldots,\mu_T,r_1,\ldots,r_h,\ldots,r_T) \tag{5.6}$$

is the PMF of a sum $Y_i = X_{i1} + \ldots + X_{in_i}$ of n_i independent random variables with

$$X_{ij_i} \sim \begin{cases} \text{NB}(\mu_1, r_1) & \text{if } 1 \leq j_i \leq J_1 \\ \vdots & \\ \text{NB}(\mu_h, r_h) & \text{if } J_{h-1} < j_i \leq J_h \\ \vdots & \\ \text{NB}(\mu_T, r_T) & \text{if } J_{T-1} < j_i \leq J_T = n_i, \end{cases}$$

with $J_1 = \ell_1, \ldots, J_h = \ell_1 + \ell_2 + \ldots + \ell_h, \ldots, J_T = \ell_1 + \ldots + \ell_T = n_i$. $f^{\text{NBNB}}_{(\ell_1,\ldots,\ell_h,\ldots,\ell_T)}$ can be calculated using the convolution of NBs described in Example 3.10. Before using this, the computation of the PMF can be simplified further: Within each population h, the distribution parameters μ_h and r_h are identical and can be transformed to the NB-parameter setting involving the `size` parameter r_h and `prob` parameter p_h. Then it follows as shown in Example 3.8 that the sum of the ℓ_j random variables follows the $\text{NB}(\ell_j r_h, p_h)$ distribution. Consequently, $f_{(\ell_1,\ldots,\ell_T)}$ is the convolution of at maximum T different NB distributions (exactly T-fold if all $\ell_j > 0$). These can than be calculated as shown in Example 3.10. In practice, Equation (3.4) is not easy to calculate as it contains an infinite sum. Therefore, we need to cut this and stop the calculation as soon as the following summands equal zero. The approximation of the density of the convoluted NBs is implemented in C++ and can be used via `d_snb()`. The overall PMF given in (5.5) of the NB-NB model, $f^{\text{NB-NB}}_{n_i}(y_i|\mu_1,\ldots,\mu_h,\ldots,\mu_T,r_1,\ldots,r_h,\ldots,r_T,p_1,\ldots,p_h,\ldots,p_T)$, with $p_1 + \cdots + p_T = 1$ was recently implemented in `d.sum.of.mixtures.NBNB()`.

Example 5.2 (Mixture of two populations - Part 2) *We continue with Example 5.1, where we suppose that the sample contains two populations. If each observation consist of the same number of $n = 10$ cells, Y_i are 10-fold convolutions for all i, and the PDF/PMF (5.5) simplifies to*

$$f_{10}\left(y_i|\boldsymbol{\theta},\boldsymbol{p}\right) = \sum_{\ell=0}^{10} \binom{10}{\ell} p^{\ell}(1-p)^{10-\ell} f_{(\ell,10-\ell)}\left(y_i|\boldsymbol{\theta}\right), \tag{5.7}$$

where $f_{(\ell,10-\ell)}$ are the specific PDF/PMF of the sum Y_i of ten independent random variables, i. e. $Y_i = X_{i1} + \ldots + X_{i10}$, where we know how many summands come from each of the two populations. This PDF/PMF depends on the particular chosen model, which are given by

LN-LN:

$$f_{(\ell,10-\ell)}(y_i|\boldsymbol{\theta}) = f^{LN\text{-}LN}_{(\ell,10-\ell)}(y_i|\mu_1,\mu_2,\sigma^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{i10}$ of ten independent random variables with

$$X_{ij} \sim \begin{cases} \text{LN}(\mu_1, \sigma^2) & \text{if } 1 \leq j \leq \ell \\ \text{LN}(\mu_2, \sigma^2) & \text{if } \ell < j \leq 10. \end{cases}$$

rLN-LN:

$$f_{(\ell,10-\ell)}(y_i|\boldsymbol{\theta}) = f^{\textit{rLN-LN}}_{(\ell,10-\ell)}(y_i|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{i10}$ *of ten independent random variables with*

$$X_{ij} \sim \begin{cases} \mathrm{LN}(\mu_1,\sigma_1^2) & \textit{if } 1 \le j \le \ell \\ \mathrm{LN}(\mu_2,\sigma_2^2) & \textit{if } \ell < j \le 10. \end{cases}$$

EXP-LN:

$$f_{(\ell,10-\ell)}(y_i|\boldsymbol{\theta}) = f^{\textit{EXP-LN}}_{(\ell,10-\ell)}(y_i|\lambda,\mu,\sigma^2)$$

is the PDF of a sum $Y_i = X_{i1} + \ldots + X_{i10}$ *of ten independent random variables with*

$$X_{ij} \sim \begin{cases} \mathrm{LN}(\mu,\sigma^2) & \textit{if } 1 \le j \le \ell \\ \mathrm{EXP}(\lambda) & \textit{if } \ell < j \le 10. \end{cases}$$

NB-NB:

$$f_{(\ell,10-\ell)}(y_i|\boldsymbol{\theta}) = f^{\textit{NB-NB}}_{(\ell,10-\ell)}(y_i|\mu_1,\mu_2,r_1,r_2)$$

is the PMF of a sum $Y_i = X_{i1} + \ldots + X_{i10}$ *of ten independent random variables with*

$$X_{ij} \sim \begin{cases} \mathrm{NB}(\mu_1,r_1) & \textit{if } 1 \le j \le \ell \\ \mathrm{NB}(\mu_2,r_2) & \textit{if } \ell < j \le 10. \end{cases}$$

Depending on which of the models is chosen, $f_{(\ell,10-\ell)}\left(y_i|\boldsymbol{\theta}\right)$ *and therefore* $f_{10}\left(y_i|\boldsymbol{\theta},\boldsymbol{p}\right)$ *can be calculated with the methods described above.*

PDF of Pooled Gene Expression for Mixed Pools When estimating a gene expression model from data, one may want to verify whether the estimated model adequately describes the data. In Figure 5.14, we do this by comparing the estimated PDF to the histogram of the data and to the true PDF: The orange curve is known since we synthetically generated this data. For the blue curve, we first estimate the model parameters and then plug these in into the general model PDF. In case of a uniform pool size across all measurements, this procedure is straightforward. For a vector of pool sizes, i. e. a mix of e. g. 1-cell, 2-cell and 10-cell data, the PDF/PMF (see e. g. Figure 5.2B) is less obvious. We calculate this function as follows:

- For each cell number contained in the n-vector, calculate the PDF/PMF of the respective pool size and plug in the parameter estimates.
- Calculate the weighted sum of these PDFs/PMFs — weighted according to the times the respective pool size occurs in the n-vector.

The resulting PDF/PMF approximates the PDF/PMF of a sample where the observations are based on the pool sizes of the considered n-vector. While this PDF/PMF describes a mixture distribution with randomly drawn pool sizes (according to the weights used), we in our applications assume the pool sizes to be known for each measurement.

5.2.4 Likelihood Function

In the previous section we described the employed mathematical models of heterogeneous gene expression. In order to perform parameter inference, the likelihood function (see Sectaion 3.6) of the parameters given some data is needed.
Overall, after re-introducing the superscript (g) for measurements of genes $g = 1, \ldots, m$, Equation (5.5) results in the gene-wise PDF

$$f_{n_i}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right) =$$

$$\sum_{\ell_1=0}^{n_i}\sum_{\ell_2=0}^{n_i-\ell_1}\cdots\sum_{\ell_{T-1}=0}^{n_i-\sum_{h=1}^{T-2}\ell_h}\binom{n_i}{\ell_1, \ell_2, \ldots, \ell_T} p_1^{\ell_1}p_2^{\ell_2}\cdots p_T^{\ell_T} f_{(\ell_1,\ell_2,\ldots,\ell_T)}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}\right) \tag{5.8}$$

with model-specific choice of $f_{(\ell_1,\ell_2,\ldots,\ell_T)}$. While $\boldsymbol{n} = (n_1, \ldots, n_k)$ is considered known, we aim to infer the unknown model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(m)}, \boldsymbol{p}\}$. Assuming independent observations $\boldsymbol{y} = \{y_i^{(g)}|i = 1, \ldots, k; g = 1, \ldots, m\}$ of $Y_i^{(g)}$ for m genes and k tissue samples, where sample i contains n_i cells, the likelihood function is given by

$$L(\boldsymbol{\theta}|\boldsymbol{y}) = \prod_{g=1}^{m}\prod_{i=1}^{k} f_{n_i}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right).$$

Consequently, the log-likelihood function (Equation 3.7) of the **stochprofML** model parameters reads

$$\ell(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{g=1}^{m}\sum_{i=1}^{k} \log\left[f_{n_i}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right)\right]. \tag{5.9}$$

Example 5.3 (Mixture of two populations - Part 3) *Continuing with Example 5.2 where the 10-cell measurements originate from a two population mixture, the log-likelihood for* $k = 100$ *tissue samples and* $m = 5$ *genes from is given by*

$$\ell(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{g=1}^{5}\sum_{i=1}^{100} \log\left[f_{10}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right)\right],$$

where $f_{10}\left(y_i^{(g)}|\boldsymbol{\theta}^{(g)}, \boldsymbol{p}\right)$ *is given by Equation* (5.7) *and depends on the chosen distribution model.*

5.3 Maximum Likelihood Estimation and Model Selection

The **stochprofML** algorithm aims to infer the unknown model parameters using maximum likelihood estimation (see Section 3.3.1). As input, an $m \times k$ data matrix of pooled gene expression, known cell numbers $\vec{n}$, the assumed number of populations T and the choice of single-cell distribution (LN-LN, rLN-LN, EXP-LN,NB-NB) is expected. Based on this input, the algorithm aims to find parameter values of $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(m)}, \boldsymbol{p}\}$ that maximize $\ell(\boldsymbol{\theta}|\boldsymbol{y})$ as given by Equation (5.9). Here we describe practical aspects of the implemented optimization procedure. As it is very complex describing this in general, we go back to the example of 10-cell measurements of a two population mixture.

Example 5.4 (Mixture of two populations - Part 4) *Several challenges occur during parameter estimation. We explain these on the two-population LN-LN example: First, one needs to ensure parameter identifiability. This is achieved for the two-population LN-LN model by constraining the parameters to fulfill either* $p \le 0.5$ *or* $\mu_1 > \mu_2$. *Otherwise, the two combinations* $(p, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \sigma)$ *and* $(1-p, \boldsymbol{\mu}_2, \boldsymbol{\mu}_1, \sigma)$ *would yield identical values of the likelihood function and could cause computational problems. For this implementation, the second possibility was preferred, i. e.* $\mu_1 > \mu_2$. *The alternative, i. e. requiring* $p \le 0.5$, *led to switchings between* μ_1 *and* μ_2 *in case of* $p \approx 0.5$. *As a second measure, we implement unconstrained rather than constrained optimization: Instead of estimating* $(p, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \sigma)$ *under the constraints* $p \in [0,1]$, $\mu_1 > \mu_2$ *and* $\sigma > 0$, *the parameters are transformed to* $(\mathit{logit}(p), \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \log(\sigma))$, *and an unconstrained optimization method is used. This is substantially faster. In detail* p *is transformed to*

$$w = \mathit{logit}(p) = \log\left(\frac{p}{1-p}\right) \in \mathbb{R}$$

and later back-transformed via

$$p = \mathit{logit}^{-1}(w) = \mathit{expit}(w) = \frac{\exp(w)}{1+\exp(w)} \in [0,1] \ .$$

The aforementioned transformations are likewise employed for all other models (rLN-LN and EXP-LN, NB-NB) and population numbers. In particular, in the continuous models σ *and* λ *are log-transformed, and the lognormal populations are ordered according to the log-means* $\mu_h^{(1)}$ *of the first gene in the gene list. In the discrete model, all population parameters –* μ *and* r *– are log-transformed analogously. The NB populations are ordered according to the means* $\mu_h^{(1)}$ *of the first gene in the gene list. To allow unconstrained optimization, the population probabilities are transformed to* $\mathbb{R}$ *in all models as described for the LN-LN model above.*

All transformations of population parameters are analogously applied to the general case, where $T > 2$. However, transforming the probabilities $p_1, \dots, p_T$ to the unrestricted space, has to be adapted to fulfill $p_h \in [0,1]$ for all $h = 1, \dots, T$ and $\sum_{h=1}^{T} p_h = 1$ after back-transformation. Therefore, we set $\tilde{p}_h = p_1 + \dots + p_h$ and use the following transformations

$$w_h = \text{logit}\left(\frac{p_1 + \dots + p_h}{p_1 + \dots + p_{h+1}}\right) = \text{logit}\left(\frac{\tilde{p}_h}{\tilde{p}_{h+1}}\right) \in \mathbb{R} \qquad \text{for all } h \in 1, \dots, T-1.$$

For the back-transformations, start at $h = T-1$ and calculate

$$\tilde{p}_h = \text{expit}(w_h)\, \tilde{p}_{h+1} \in [0,1] \qquad \text{for all } h \in T-1, \dots, 1$$

in reverse order. Setting $\tilde{p}_T = 1$ ensures that the probabilities sum up to one. Additionally, one has $\tilde{p}_h \leq \tilde{p}_{h+1}$ as $\text{expit}(w_h) \in [0,1]$ for all $h \in 1, \ldots, T-1$. Obviously, $p_1 = \tilde{p}_1$, and the remaining population probabilities are given by

$$p_h = \tilde{p}_h - \tilde{p}_{h-1} \in [0,1] \qquad \text{for all } h \in 2, \ldots, T.$$

The log-likelihood function is multimodal. Thus, a single application of some gradient-based optimization method does not suffice to find a global maximum. Instead, two approaches are combined which are alternately executed: First, a grid search is performed, where the log-likelihood function is computed at many randomly drawn parameter values. In the second step, the (computationally more costly) Nelder-Mead algorithm Nelder and Mead (1965) is repeatedly executed at few points. This way, high likelihood regions can be identified with low computational cost. A next grid search again explores the regions around the obtained local maxima, followed by another Nelder-Mead optimization. Here, the starting values are randomly drawn from the high-likelihood regions found before. This combination of grid search and local optimization is carried out three times. The whole procedure is repeated five times by default, with the aim to find an overall optimal parameter combination, but this number can be changed or can be stopped early as soon as the algorithm has converged, this is when the improvement in the likelihood during the last round is less than $5 \cdot 10^{-5}$.

If a dataset contains gene expressions for m genes, and T populations are assumed, there are at minimum $T(m+1)$ parameters which one seeks to estimate depending on the model framework. This is computationally difficult, because the number of modes of the log-likelihood function increases with the number of parameters. The performance of the numerical optimization crucially depends on the quality of the starting values, and a large number of restarts is required. When analyzing a large gene cluster, it is advantageous to start by considering small clusters and use the derived estimates as initial guesses for larger clusters. Approximate marginal 95% confidence intervals for the parameter estimates are obtained as follows: We numerically compute the Hessian matrix of the negative log-likelihood function on the unrestricted parameter space and evaluate it at the (transformed) maximum likelihood estimator. Denote by d_i the ith diagonal element of the inverse of this matrix. Then the confidence bounds for the ith transformed parameter θ_i are

$$\hat{\theta}_i \pm 1.96\sqrt{d_i}.$$

We obtain respective marginal confidence intervals for the original true parameters by back-transformation of the above bounds. This approximation is especially appropriate in the two-population LN-LN example for the parameters p and σ when conditioning on $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. In this case, in practice, the profile likelihood is seemingly unimodal. In Appendix F examples are shown how to use the **stochprofML** package. Run times for maximum likelihood estimation differ substantially between two- and three-population models, and also between LN-LN, rLN-LN and EXP-LN. The latter is due to the integral convolution of an exponential and an Erlang distribution in EXP-LN as described above. Table 5.1 displays run times using the R function `microbenchmark()` on simulated data.

T	LN-LN	rLN-LN	EXP-LN
2	13.00 (12.55 - 18.99)	27.06 (17.04 - 34.76)	16,762.22 (10,764.77 - 21,576.25)
3	96.76 (47.07 - 130.92)	162.59 (86.84 - 346.98)	160,391.65 (117,985.89 - 186,557.13)

Table 5.1: Run times for maximum likelihood estimation for LN-LN, rLN-LN and EXP-LN models with $T = 2$ and $T = 3$ populations. The study was performed on simulated data using the R function microbenchmark(). Reported numbers are run times in seconds across five repetitions: median (min - max).

Example: Mixture of three populations Figure 5.3 shows estimation results for an LN-LN model with three populations, based on synthetic 10-cell data. (Synthetic data generation is described later in this text.) 1,000 10-cell datasets each with $k = 1,000$ observations were generated using underlying population parameters $p_1 = 0.1$, $p_2 = 0.4$, $\mu_1 = 1.5$, $\mu_2 = -0.4$, $\mu_3 = -2.5$ and $\sigma = 0.2$.

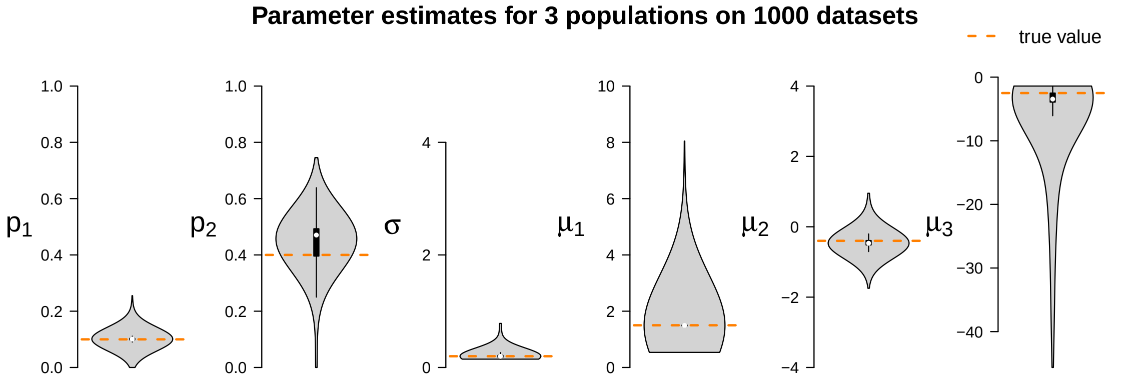


Figure 5.3: Parameter estimates for the LN-LN model on 1,000 simulated 10-cell datasets. The true underlying population parameters are $p_1 = 0.1$, $p_2 = 0.4$, $\mu_1 = 1.5$, $\mu_2 = -0.4$, $\mu_3 = -2.5$ and $\sigma = 0.2$, as indicated by the orange dashed lines.

As described above, the number of populations is not determined during parameter inference. Instead parameter inference has to be performed for several population numbers and then the model that fitted the data best has to be selected. In general by increasing the number T of populations, the observed data can be modeled more precisely, but this comes at the cost of potential overfitting. For example, a three-population LN-LN model may lead to a larger likelihood at the maximum likelihood estimator than a two-population LN-LN model on the same dataset. However, the difference may be small, and the additional third population may not lead to a gain of knowledge. For example, the estimated population probability $\hat{p}_3$ may be tiny, or the log-means of the second and third population, $\hat{\mu}_2$ and $\hat{\mu}_3$ might hardly be distinguishable from each other.

To objectively find a trade-off between necessary complexity and sufficient interpretability, we employ the Bayesian information criterion (BIC, see 3.9) that includes maximum likelihood estimate of the respective model, the number of parameters and the size of the dataset.

In practice, it is required to estimate all models of interest separately with the **stochprofML** algorithm, e. g. the LN-LN model with one, two and three populations, and/or the respective rLN-LN and EXP-LN models. For discrete models the NB-NB model can be fitted with different number of populations. The BIC values are returned by the function `stochprof.loop()`.

5.4 Simulation Studies Using the stochprofML Package

Next we demonstrate the performance of the maximum likelihood estimation via the **stochprofML** package depending on pool sizes (Section 5.4.1), true parameter values (Section 5.4.2) and in case of uncertainty about pool sizes (Section 5.4.3). These investigations shed light on the algorithm's performance from a statistical point of view and complement the experimental validation that were performed in Bajikar et al. (2014). All scripts used in these studies can be found in our open GitHub repository `https://github.com/fuchslab/Stochastic_Profiling_in_R`. The general procedure in the following simulation studies is to first generate synthetic datasets with some predefined population parameters and frequencies using `r.sum.of.mixtures()`. Thereby datasets with either fixed or varying pool sizes are generated, i. e. the numbers of cells contained in one pool are fixed or vary from cell pool to cell pool within a dataset. Next, we assume that we do not know the predefined model parameters and estimate them using `stochprof.loop()`. Using simple summary statistics, we compare the estimates of the parameters in different ways, e. g. how they are influenced by increasing cell numbers or how their variance differs when the dataset was generated with differing population parameters.

First, we give an overview about the different model parameter settings and pool sizes used in data generation: We use datasets with fixed pool sizes that contain single-cells, 2 cells, 5 cells, 10 cells, 15 cells, 20 cells or 50 cells. Additionally, we chose two types of datasets with varying pool sizes. The first contains small cell pools with 1, 2, 5 and 10 cells, the second contains larger cell pools with 10, 15, 20 and 50 cells. Thus, in total we have nine different cell pool settings that we use for data generation.

In all simulation studies, we use the LN-LN model with the five different parameter settings, given in Table 5.2. While the first set is considered to be the default, each of the other parameter sets differs from it in one of the population parameters. Taken together, for each of the nine cell pool settings and each of the five parameter settings 1,000 datasets are generated using `r.sum.of.mixtures.LNLN()`, so that in total we have generated `5*9*1000` = 4.5×10^4 datasets.

5.4.1 Simulation Study on Optimal Pool Size

Stochastic profiling, i.e. the analysis of small-pool gene expression measurements, is a compromise between the analysis of single cells and the consideration of large bulks:

	p	μ_1	μ_2	σ
Set 1	0.2	2	0	0.2
Set 2	0.1	2	0	0.2
Set 3	0.4	2	0	0.2
Set 4	0.2	2	1	0.2
Set 5	0.2	2	0	0.5

Table 5.2: Overview of the five model parameter settings used in the 5.4.1 and in the 5.4.2.

Single-cell information is most immediate, but a fixed number k of samples will only cover k cells. In pools of cells, on the other hand, information is convoluted, but k pools of size n cover n times as much material. An obvious question is the optimal pool size n. The answer is not available in analytically closed form. We hence study this question empirically.

As described above, we generate synthetic data for different pool sizes with identical parameter values and settings. Then, we re-infer the model parameters using the **stochprofML** algorithm. This is repeated $1,000$ times for each choice of pool size, enabling us to study the algorithm's performance by simple summary statistics of the replicates.

Figure 5.4 summarizes the point estimates of the 1,000 datasets for each of the nine pool size settings generated with parameter set 1. It seems that (for this particular choice of model parameter values) parameter estimation works reliably for pool sizes up to ten cells, with smaller variance from single-cells to 5-cells. This applies also for the mixture of pool sizes for the small cell numbers. For cell numbers larger than ten, the range of estimated values becomes considerably larger, but without obvious bias, which also applies to the mixture of the larger pool sizes.

Figure 5.4 suggests $n = 5$ or varying small pool sizes as ideal choices since its estimates show smaller variance than the other pool sizes. This simulation study, however, has been performed in an idealized in silico setting: We did not include any measurement noise. In practice, however, it is well known that single-cells suffer more from such noise than samples with many cells. The ideal choice of pool size may hence be larger in practice.

Appendix G shows the figures of the repetitions of this study for the other four sets of population parameters. The results there confirm the observations just made.

In the second parameter setting, the fraction of the first population was reduced to 10% as compared to the first parameter setting. The results are shown in Figure G.1. They are similar to the results of the first parameter set in Figure 5.4. For set 2, however, single cells lead to large variance of estimates, supposedly due to the small sample size of 50 in combination with the small probability (10%) of the first population: We can only expect five single cells of the first population to be measured on average. In some datasets, this will be too low to estimate the parameters of

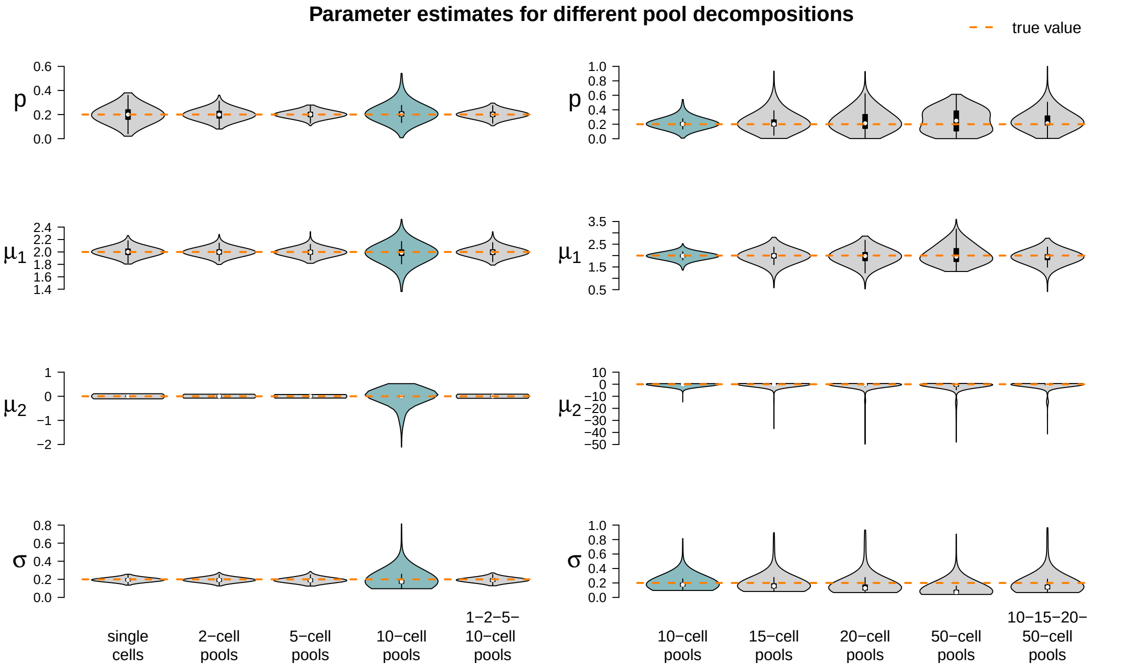


Figure 5.4: Violin plots of parameter estimates for two-population LN-LN model on 9,000 simulated datasets, i. e. on 1,000 datasets for each pool size composition. *Left:* Results for single-cell, 2-cell, 5-cell, 10-cell pool and their mixture. *Right:* Results for larger pool sizes, namely 10-, 15-, 20-, 50-cell pools and their mixture. *Turquoise:* Results for 10-cell pools; these are repeated across the left and right panels. The true parameters are marked in orange.

the first population and/or their proportion satisfactorily. Consequently, the violins of the single-cell estimates show a higher variance, especially for the estimates of the parameters of the first population. In the third parameter setting, the fraction of the first population was increased to 40%. The resulting estimates are shown in Figure G.2. In this setting, both populations are similarly frequent; hence, it seems plausible that the single-cell estimates show similar variability as for example the 2-cell estimates. The estimates of the mixed pools of the lower cell numbers provide estimates that are as accurate as the ones for single-cell and 2-cell data. From a pool size of five cells on, the estimates vary strongly. Apparently, low cell numbers are advisable if a tissue is not dominated by one cell population. In the fourth parameter setting, μ_2 is increased to 1 and thus larger than in the first parameter setting. The two populations are more similar. The resulting estimates are shown in Figure G.3. Starting from a pool size of 10 cells, it seems as if the variance of the estimates did not increase any more. The estimates for the mixed pools with larger cell numbers can sometimes not distinguish the populations, therefore the violin of p is bi-modal. We draw the same conclusion as for two populations with similar frequencies that more similar populations should be investigated in pools with lower cell numbers because their individual expression profile is blurred for small pool sizes already. Finally, we investigate the effect of different pool sizes in the fifth parameter set, where the log-sd σ of both populations is increased to 0.5. The resulting estimates of the model parameters are shown in Figure G.4. With an increase of σ, both populations have broader distributions. It appears that there is an increase in variance in the estimates between the 5-cell and the 10-cell measurements. Increasing cell numbers in the pools mainly influences the estimate of σ, which is increasingly underestimated.

5.4.2 Simulation Study on Impact of Parameter Values

The underlying data-generating model obviously influences the ability of the maximum likelihood estimator to re-infer the true parameter values: Values of p_1 close to 0.5, small differences between μ_1 and μ_2 and large σ blur the data and complicate parameter inference in practice. In the next simulation study, we investigate the sensitivity of parameter inference and which scenarios could be realistically identified. We use the same datasets as in the previous simulation study: The parameter choices from set 1 are considered as the standard and compared to the other four settings. In detail, p_1 is reduced from 0.2 to 0.1 in one setting and increased to 0.4 in the next. μ_2 is increased from 0 to 1, and σ increases from 0.2 to 0.5. μ_1 is kept fixed to 2 in all settings. As before, we consider 1,000 data sets for every parameter setting and compare the resulting estimates to the true values.

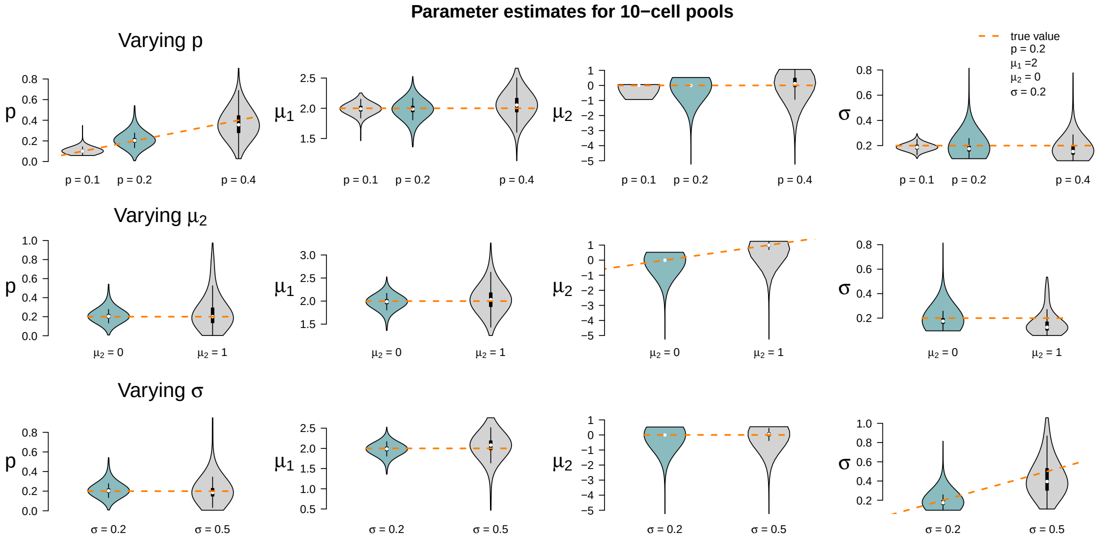


Figure 5.5: Violin plots of parameter estimates for two-population LN-LN model for varying parameters p, μ_2 and σ. Five parameter sets (see Appendix G) were used to simulate 1,000 datasets from each of which they were back-inferred. Violin plots for the standard setting $p = 0.2$, $\mu_1 = 2$, $\mu_2 = 0$ and $\sigma = 0.2$ are colored turquoise. The true parameters used to simulate the data are marked in orange.

Figure 5.5 shows the results of the study. In each row of the plot, we compare the estimates of the datasets that were simulated with the standard parameters to the estimates of the datasets that were simulated with one of the parameters changed. Even if only one parameter is changed all parameters are estimated. Each violin accumulates the estimates of 1,000 datasets. For easier comparison, each of the twelve tiles shows the standard setting as turquoise violin, which means those are repeated in each row.

When changing the parameter values, they can still be derived without obvious additional bias, but accuracy decreases for increasing p, decreasing $\mu_2 - \mu_1$ and increasing σ (with few exceptions).

Appendix G contains the remaining figures of the other pool sizes. Results for single-cell and 2-cell pools look alike (Figures G.5 and G.6). As discussed before, the variance of the estimates become large for a small value of p in combination with the small pool sizes. For both single-cell and 2-cell data, varying μ_2 does not affect the estimation accuracy of the estimation, whereas a larger value of σ leads to higher variance of all parameter estimates but for p. In contrast to this, the 5-cell data results in a different pattern (Figure G.7): As compared to the estimates from the standard setting, the estimates show a larger variance. The mixture of small cell pool numbers (Figure G.8), however, lead to similar results as the pure 2-cell datasets. Figure G.9 displays the results for the 15-cell data. For most parameter combinations, the variance of the estimates does not change dramatically. The most accurate estimates are achieved for small p, the least accurate ones for large σ, in which case σ gets underestimated. The same holds true for the 20- and 50-cell datasets (Figures G.10 and G.11), with even larger variance. For the mixture of large cell pools (Figure G.12), estimation performance is comparable to the one for the pure 50-cell measurements.

Taken together, the result for other pool sizes show that the observations made on the 10-cell pools can be transferred to other pool sizes with some additions: Larger pool sizes infer parameters more accurately if p is smaller. In an increased first population setting ($p = 40\%$), μ_1 can be better inferred if the data set consists of smaller pools. For larger pools, the estimation of μ_1 and μ_2 works comparably well after increasing μ_2. In general, the estimation of σ is the most difficult one: As shown in Equation (A.1), the mean (and variance) of the lognormal distribution is determined by both the parameters μ_1 and μ_2 and by σ. Estimates of σ will be negatively correlated with estimates $\hat{\mu}_1$ and $\hat{\mu}_2$ if the mean is determined correctly. Indeed, in pools of 15 cells with increased σ, we see that μ_1 is slightly overestimated. Therefore, to keep the mean σ is underestimated. This worsens in larger pools.

5.4.3 Simulation Study on the Uncertainty of Pool Sizes

One key assumption of the **stochprofML** algorithm is that the exact number of cells in each cell pool is known. In Janes et al. (2010), accordingly, ten cells were randomly taken from each sample by experimental design. However, different experimental protocols may not reveal the exact cell number: In Tirier et al. (2019), for example, tissue samples were taken as whole cancer spheroids. Here, the cell numbers were experimentally unknown but estimated using light sheet microscopy and 3D image analysis. Since the **stochprofML** algorithm requires the pool sizes as input parameter, some estimate has to be passed to it. It is intuitively obvious that the better the prior knowledge about the cell pool sizes, the better the final model parameter estimate.

In this simulation study, we investigate the consequences of misspecification.

In the first part of this simulation study, we reuse the 1,000 synthetic 10-cell datasets from Section 5.4.1. Each of these contains 50 10-cell samples, simulated with underlying model parameters $p = 0.2$, $\mu_1 = 2$, $\mu_2 = 0$ and $\sigma = 0.2$. As before, we re-infer the population parameters using the **stochprofML** algorithm. This time, however, we use varying pool sizes from 5 to 15 as input parameters of the algorithm. This is a misspecification except for the true value 10. The resulting parameter estimates (empirical median and 2.5%-/97.5%-quantiles across the 1,000 datasets) are depicted in Figure 5.6.

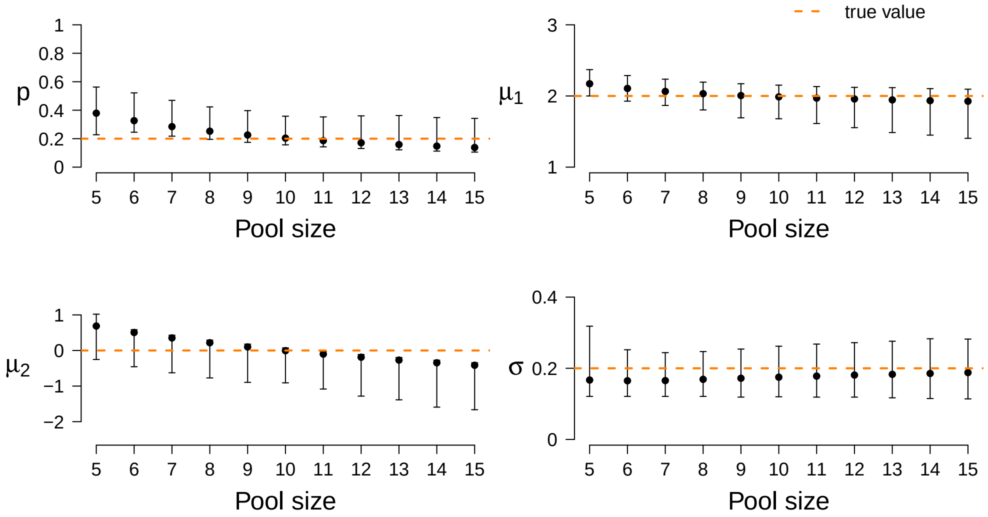


Figure 5.6: Parameter estimates for (partly) misspecified pool sizes across 1,000 synthetic datasets: The true pool size is 10 in every dataset. The **stochprofML** algorithm, however, uses values from 5 to 15 as input parameter. Bars cover the range between the empirical 2.5%- and 97.5%-quantiles. The dots mark the empirical median, the orange line the true parameter values used for simulation.

Estimates are optimal or at least among the best in terms of empirical bias and variance when using the correct pool size. With increasing assumed cell number, the estimates of p decrease, i. e. the fraction of cells from the higher expressed population is assumed to be smaller. This is a reasonable consequence of overestimating n, because in this case the surplus cells are assigned to the second population with lower (or even close-to-zero) expression. Consequently, at the same time the estimates of μ_2 decrease to be even smaller.

In the second part of this simulation study, we use the two settings with mixed cell pool sizes as introduced in Section 5.4.1. One setting embraces cell pools with rather small cell numbers (single-, 2-, 5- and 10-cell samples), the other one pools with larger cell numbers (10-, 15-, 20- and 50-cell samples). For each of the two scenarios, we generate one dataset with 50 samples. We denote the true 50-dimensional pool size vectors by $\vec{n}_{\text{small}}$ and $\vec{n}_{\text{large}}$ and employ these vectors for re-estimating the model parameters p, μ_1, μ_2 and σ. Then, we estimate the parameters again for the same two datasets for 1,000 times, but this time using perturbed pool size vectors as input to the algorithm, introducing artificial misspecification. These 50-dimensional pool size vectors are generated as follows: For each component, we draw a Poisson-distributed random variable with intensity parameter equal to the respective component of the true vectors $\vec{n}_{\text{small}}$ or $\vec{n}_{\text{large}}$. Zeros are set to one, the minimum pool size. Figure 5.7 shows these $2 \times 1,000$ parameter estimates as compared to the true parameter values and those for which the true size vectors $\vec{n}_{\text{small}}$ and $\vec{n}_{\text{large}}$ were used as input.

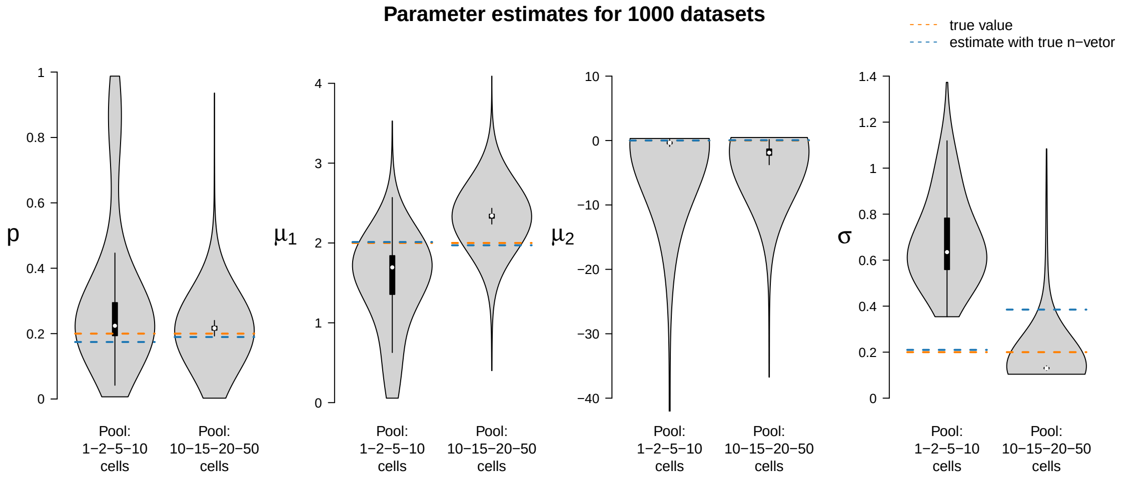


Figure 5.7: Parameter inference under misspecification of the cell pool size: Parameters are estimated for two datasets, one generated based on a pool size vector $\vec{n}_{\text{small}}$ with values between 1 and 10 (*left violin in each panel*); the other one based on a vector $\vec{n}_{\text{large}}$ with values between 10 and 50 (*right violin in each panel*). *From left to right:* Estimates of p, μ_1, μ_2 and σ. The violins depict estimates across 1,000 estimation runs, where each relies on a randomly sampled misspecified pool size vector as described in the main text. *Orange:* True parameters values. *Light blue:* Estimates without misspecification of the pool size vector.

The violins of the estimates for the smaller cell pools (based on $\vec{n}_{\text{small}}$) indicate that the estimates of p and μ_1 are fairly accurate, but the estimates of μ_2 have large variance, and σ is overestimated in all 1,000 runs. This is plausible as population 1 (the one with higher log-mean gene expression) is only present on average in 20% of the cells; even when misspecifying the pool sizes, the cells of population 1 are still detectable since this is the population responsible for most gene expression. Consequently, all remaining cells are assigned to population 2, which has lower or even almost no expression. If the pool size is assumed too low, this second population will be estimated to have on average a higher expression; if it is assumed too large, the second population will be estimated to have a lower expression. This leads to a broader distribution and thus an overestimation of σ.

The results for the larger cell pools (based on $\vec{n}_{\text{large}}$) show a similar pattern. In this case, however, the impact of misspecification is less visible, as also confirmed by additional simulations in Appendix G. For large cell pools, the averaging effect across cells is strong anyway and in that sense more robust. In the study here, due to variability of parameter estimates, the σ parameter is often even better estimated when using a misspecified pool size vector than when using the true one. It might also be appropriate to repeat the parameter estimation, as shown here, with similar pool size vectors to get more robust estimates.

Taken together, **stochprofML** can be used even if exact pool sizes are unknown. In that case, the numbers should be approximated as well as possible.

5.5 Bayesian Parameter Inference

In all models presented in this thesis, we assume that gene expression is stochastic and therefore contains large parts of variability. This variability is not only found in the data, but is also part of the rate parameters that control gene expression. Using the classical maximum likelihood inference as described above, this variability of parameters is not taken into account, since only point estimates of the parameters are returned.

Bayesian parameter estimation (Section 3.3.2) enables us to estimate the variability of the parameters by assuming that they are not fixed values but follow a distribution. Kurz (2015) proposed a Bayesian extension for the LN-LN model. Here, we will present an approach that concentrates on the discrete NB-NB model.

For this, we use the HMC-based No-U-Turn sampler (NUTS, Hoffman and Gelman, 2014) implemented in the programming language Stan through its interface **RStan** (Stan Development Team, 2019) to estimate parameters $\boldsymbol{\theta}$ and $\boldsymbol{p}$ of Equation (5.8). More information can be found in Section 3.3.2. Remember that in the NB-NB model, $\boldsymbol{\theta}$ is the parameter vector of all T involved NB distributions, defined by μ and r i. e. $\boldsymbol{\theta} = (\theta_1, \dots, \theta_T) = ((\mu_1, r_1), \dots, (\mu_T, r_T))$. In the following, we will introduce two different implementations of our model, i. e. the sum of NB mixtures in Stan.

5.5.1 Implementation of the Likelihood Function in Stan

We use the same model as in the NB-NB version of the **stochprofML** algorithm, described in Section 5.2 i. e.

$$Y_i = X_{i1} + \cdots + X_{in},$$

where n is the number of cells in each measurement and $i \in \{1, \dots, k\}$ are the observations. Now,

$$X_{ij} \sim p_1 \text{NB}\left(\alpha_1, \frac{\beta_1}{\beta_1 + 1}\right) + \cdots + p_T \text{NB}\left(\alpha_T, \frac{\beta_T}{\beta_T + 1}\right),$$

for $i = 1, \dots, k$ and $j = 1, \dots, n$ are the latent single-cell observations with $\sum_{h=1}^{T} p_h = 1$ and $\text{NB}(r, p)$ is given in Definition A.8. Note that in the following, we use the NB distribution parameterized by α and β as shown in Example 3.12.

When using RStan, the model needs to be written in Stan language. There, we cannot use our implementation of the density function $f_{(\ell_1, \ell_2, \dots, \ell_T)}\left(y_i^{(g)} | \boldsymbol{\theta}^{(g)}\right)$ – the PMF of a sum of n_i NBs as described in Equation (5.6) – since we cut the infinite sum at different points as soon as the subsequent summands are zero. This cutting point depends on the parameters and the given data.

The NUTS requires calculating the gradient of the log-posterior density. For this purpose, Stan uses auto-differentiation and creates a so-called expression tree to evaluate all required gradients of the likelihood. Since this cutting point is parameter dependent, but the expression graph is built only once at the beginning (independent of parameter inputs), the auto differentiation fails. This is because the number of summands cannot vary for each iteration (i. e. for a new parameter proposal) since the size of the expression tree would vary but can only be fix. Therefore, we need to re-implement the density function and with this the complete likelihood function in Stan code without such a varying cutting point. One way is to always approximate the sum by a constant very high number of summands, e. g. 10,000. We use an alternative solution, that implements different versions with different constant numbers of summands (i. e. 1, 5, 10, 50, 100, 500, 1,000, 5,000 and 10,000). Then one expression tree can be built with several subtrees (in this case: nine), and in each iteration, it is checked whether more summands are needed and thus which subtree to use. We call this Stan model the NB implementation.

We apply the NB implementation to a synthetic dataset with $1,000$ 2-cell samples of two populations and frequencies $\boldsymbol{p} = (0.2, 0.8)$ and NB parameters $\boldsymbol{\alpha} = (20, 70)$ and $\boldsymbol{\beta} = \left(\frac{0.1}{1-0.1} = 0.111, \frac{0.4}{1-0.4} = 0.667\right)$.

Figure 5.8 shows the chains and density plots of the resulting run. It indicates that our algorithm is able to capture the true parameter values.

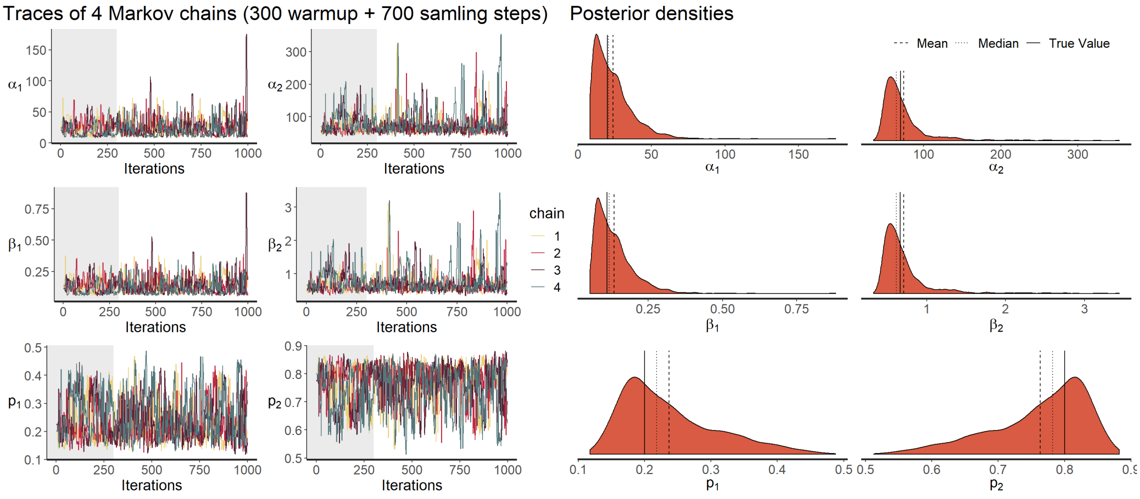


Figure 5.8: Parameter traces and densities of the posterior sample obtained with the No-U-Turn sampler (NUTS) using the NB implementation.

It is worth mentioning that running this Stan model does not return any errors, i. e. no divergences, tree depth was not exceeded and the estimated Bayesian Fraction of Missing Information (BFMI) indicates no pathological behavior. Additionally, the effective sample size n_{eff} is larger than 100 and $\hat{R}$ close to one, which tells us that

chains have mixed and the posterior distribution of the parameters were adequately estimated.
However, as shown in Table 5.3 the runtime of all four chains is three to four weeks.

	Chain 1	Chain 2	Chain 3	Chain 4	Mean
Warmup time in s	563,376	521,653	568,014	884,648	634,423
Sample time in s	1,161,320	1,233,040	1,299,130	1,476,770	1,292,565
Total time in days	19.96	20.31	21.61	27.33	22.30

Table 5.3: Runtimes of the four chains (shown in Figure 5.8) of the NUTS using the NB implementation.

Modifying this implementation to approximate Formula (3.4) more efficiently and to decrease its evaluation time will be the topic of the next section.

5.5.2 Alternative Model: Using the Poisson-Gamma Distribution

The previous section showed that the implemented Stan model works but is very time consuming (see Table 5.3 with runtimes between 20 and 27 days). Calculating more complicated models with more cells and populations would take even longer. Therefore, we search for an alternative parametrization. To additionally circumvent the problem of cutting the infinite sum and creating huge expression trees, we use the fact that a NB distribution is the same as a PG (Poisson-gamma) distribution (see Example 3.3). The hierarchical perspective of using a Poisson distribution with gamma distributed intensity parameter results in the same random variables but allows us to look at a convolution of PGs instead of a convolution of NBs. In the Bayesian world, this is a difference. By Defnition 3.5, the convolution of compound Poisson distributions results in a compound Poisson distribution where the intensity parameter is a convolution of intensity distributions, in our case Gamma distributions. Therefore, we introduce latent parameters λ that follow Gamma(α, β) distributions into the Stan model. In detail, we now use the following model:

$$Y_i = X_{i1} + \cdots + X_{in},$$

where n is the number of cells in each measurement and $i \in 1, \dots, k$ are the observations. As before

$$X_{ij} \sim p_1 \text{Pois}(\lambda_1) + \cdots + p_T \text{Pois}(\lambda_T)$$

are the latent single-cell observations with $\sum_{h=1}^{T} p_h = 1$. Additionally,

$$\lambda_h \sim \text{Gamma}(\alpha_h, \beta_h)$$

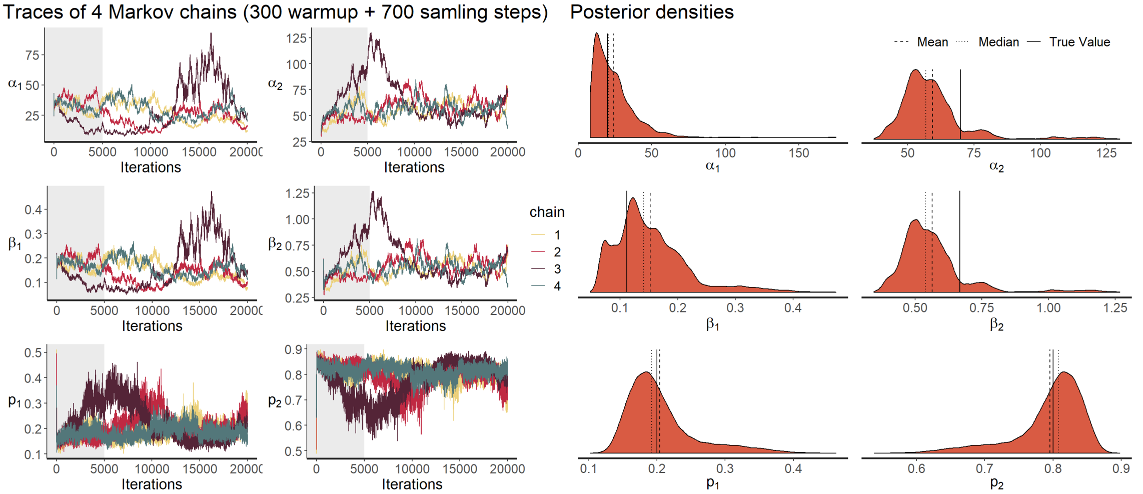
introduces a layer of latent parameters λ_h.
This model does not need the expensive implementation and calculation of Equation (3.4), and thus, the expression tree is much smaller. However, chains are sampled

for all T latent parameters λ_h. Reparametrizing them to $\bar{\lambda}_h$, where $\lambda_h = \bar{\lambda}_h/\beta_h$, leads to more stable chains for the latent parameters $\bar{\lambda}_h$ since

$$\bar{\lambda}_h \sim \text{Gamma}(\alpha_{\text{h}}, 1)$$

only depends on one parameter.

Figure 5.9 shows results from this alternative version using the hierarchical PG distribution and the reparametrization introduced above. We will call this alternative model PG implementation. In contrast to the NB implementation in Section 5.5.1, we already see in the trace plots of the PG implementation that chains did not mix very well. This can also be detected in the effective sample size n_{eff} which is very low (between 12 and 19) and $\hat{R}$ is greater than one (1.2 – 1.8). Note that we already increased chain lengths from 1,000 to 20,000 in order to get larger effective sample sizes. Although the run does not return problems in divergence or tree depth, the BFMIs of all four chains are below 0.2 indicating that we may need to reparametrize the model. We know that this is possible but comes with higher computational costs. Nevertheless, Table 5.4 shows that even with these longer chains, run times are substantially smaller (between 8 and 12 hours) than before.


Traces of 4 Markov chains (300 warmup + 700 samling steps)

Posterior densities

Figure 5.9: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation.

	Chain 1	Chain 2	Chain 3	Chain 4	Mean
Warmup time in s	4,753	4,396	6,842	4,364	5,089
Sample time in s	27,546	27,078	34,199	27,096	28,990
Total time in h	8.97	8.74	11.40	8.74	9.46

Table 5.4: Runtimes of the four chains (shown in Figure 5.9) of the NUTS using the PG implementation.

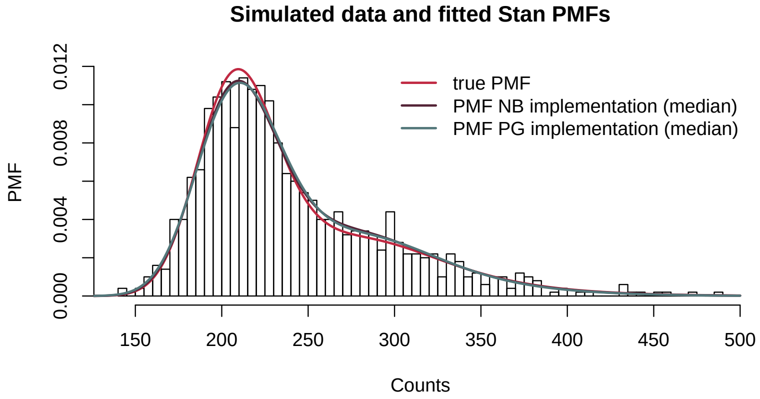


Figure 5.10: Histogram of the simulated data with the true 10-cell PMF of the original parameters in red. The resulting NB implementation is plotted in brown and the PG implementation in turquoise. For both the median of the parameter estimates is selected.

In the density plots of Figure 5.9, we can observe that the estimated population parameters are close to the original ones.
Figure 5.10 shows the histogram of the simulated data and its density estimates. The true PMF with the original parameter values is also shown. Now, we compare the fitted PMFs of the NB implementation and the PG implementation with the data. First, we see that both fitted PMFs lie nearly perfectly on top of each other, showing that the simplified, unperfect PG implementation comes to the same result. Additionally, both PMFs are close to the true and the real density of the data. This means, that the simplified PG implementation fits the 10-cell PMF comparably well to the data as the time-consuming NB implementation.
Taken together, we conclude that this alternative parametrization is not as good as the original NB parametrization. The chains look worse together with the $\hat{R}$ as well as the effective sample size n_{eff}, we conclude that chains have not mixed very well. Additionally, the low values of the BFMI suggest to reparametrize the model. On the bright side, this model takes much less time to be computed (between 8 and 12 hours with a mean of 9.45 hours) and the fits seem comparably well.
We suggest that if we want to use the Stan model, we use the simplified PG implementation. The NB implementation just takes too long and therefore is not usable. When using the PG implementation, we can still tune the step size parameter ϵ manually as well as the maximal tree size parameter.

5.5.3 stochprofML: Bayesian Inference versus ML Optimization

Next, we compare the results when fitting data with the proposed Stan model using the simplified PG implementation from the previous section with the output of the **stochprofML** algorithm using the NB-NB model. For this, we generated 8 synthetic datasets with either one homogeneous population or a two-population mixture. For each of these population compositions, we generate four datasets that contain 1,000 observations of single cells, pools of 2 cells, 5 cells or 10 cells, respectively. We applied the PG Stan model as well as the **stochprofML** algorithm using the NB-NB model to infer the population parameters.

In Figure 5.11, we show the histogram of the data and its true density together with the fitted densities of both fits. As in the previous section, we use the median for the Stan parameter fit. Appendix H contains all parameter chains and densities.

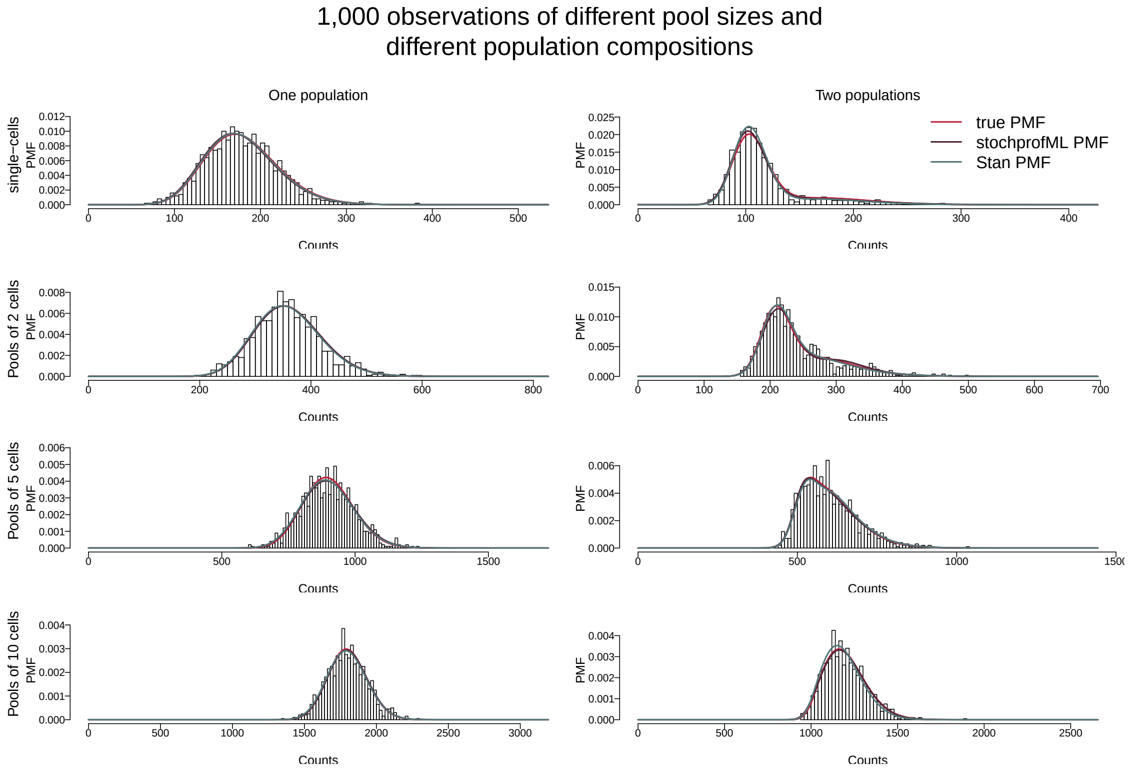


Figure 5.11: Histogram of the simulated data with the true PMF of the original parameters in red. The PMF of the **stochprofML** fit using the NB-NB model is plotted is brown and the Stan fit using the PG implementation in turquoise. For the Stan fit, the median of the parameter estimates is selected.

Table 5.5 shows the true and the estimated parameters together of the **stochprofML** fit using the NB-NB model and the Stan fit using the PG implementation with their BICs for all of the 8 simulated datasets (all values are rounded). We see that for each dataset both the **stochprofML** NB-NB model and the Stan PG implementation result in BICs very close to and often smaller than the BIC calculated for the true parameters that were used for the data simulation. Note that often the BIC of the Stan fit is much closer. In more than half of the datasets the Stan fit resulted in the smallest BIC.

Both models are very time consuming, especially for more cells and more populations. Computing times range from 8 seconds to 6.5 days for the **stochprofML** and from 56 minutes to 14 hours for the Stan runs. Since the Stan run creates comparably long chains for all model specifications, parameter inference for a small model such as one population fit on single-cell data take substantially longer than the **stochprofML**

		1 Population		2 Populations	
		NB_{μ_1,r_1}	**BIC**	$p_1\ \mathrm{NB}_{\mu_1,r_1}+p_2\ \mathrm{NB}_{\mu_2,r_2}$	**BIC**
sc	Truth	$\mathrm{NB}_{180,20}$	10,300	$0.20\ \mathrm{NB}_{180,20}+0.80\ \mathrm{NB}_{105,70}$	9,264
	stochprofML	$\mathbf{NB_{178,20}}$	**10,295**	$0.21\ \mathrm{NB}_{172,12}+0.79\ \mathrm{NB}_{104,78}$	9,458
	Stan	$\mathrm{NB}_{179,20}$	10,300	$\mathbf{0.16\ NB_{178,22}+0.84\ NB_{105,85}}$	**9,250**
2-cells	Truth	$\mathrm{NB}_{180,20}$	11,026	$0.20\ \mathrm{NB}_{180,20}+0.80\ \mathrm{NB}_{105,70}$	10,462
	stochprofML	$\mathrm{NB}_{180,20}$	11,034	$0.16\ \mathrm{NB}_{196,33}+0.84\ \mathrm{NB}_{107,56}$	10,569
	Stan	$\mathbf{NB_{179,20}}$	**11,025**	$\mathbf{0.23\ NB_{169,13}+0.77\ NB_{105,75}}$	**10,457**
5-cells	Truth	$\mathrm{NB}_{180,20}$	12,029	$0.20\ \mathrm{NB}_{180,20}+0.80\ \mathrm{NB}_{105,70}$	11,629
	stochprofML	$\mathrm{NB}_{180,20}$	12,048	$0.32\ \mathrm{NB}_{156,90}+0.68\ \mathrm{NB}_{103,14}$	11,697
	Stan	$\mathbf{NB_{180,18}}$	**12,025**	$\mathbf{0.23\ NB_{184,22}+0.77\ NB_{105,73}}$	**11,629**
10-cells	Truth	$\mathrm{NB}_{180,20}$	12,695	$0.20\ \mathrm{NB}_{180,20}+0.80\ \mathrm{NB}_{105,70}$	12,370
	stochprofML	$\mathbf{NB_{180,19}}$	**12,677**	$0.45\ \mathrm{NB}_{145,11}+0.55\ \mathrm{NB}_{99,27}$	12,435
	Stan	$\mathrm{NB}_{180,19}$	12,693	$\mathbf{0.20\ NB_{174,20}+0.80\ NB_{145,71}}$	**12,381**

Table 5.5: Parameter values and BIC of the true parameters used for data simulation, the **stochprofML** fit using the NB-NB model and the Stan fit using the PG implementation. Values are rounded. The model with the smallest BIC is printed in bold.

run. But then again computation time is greatly reduced for bigger models compared to the **stochprofML** implementation because the complex and time-consuming log-likelihood does not have to be calculated.

We conclude that both methods to infer the population parameters work in general. We need to have in mind that the Stan model does not run perfectly and we cannot be sure that samples are generated from the posterior distribution. Therefore, we cannot recommend to use the output for any posterior distribution-based investigations. Nevertheless, we can use the chains to generate a median estimate and to calculate the resulting BIC for model comparison. In Section 5.5.1, we found a good Stan model, but since it is very time consuming, it is not usable in practice.

Taken together, using the **stochprofML** package with its maximum likelihood optimization is a good suggestion, since then we are sure that all the conditions are fulfilled and we can really trust the results. Furthermore, additional model and parameter statistics can be calculated such as confidence intervals.

5.6 Interpretation of Estimated Heterogeneity

We investigate what we can learn from the parameter estimates about the heterogeneous populations (Section 5.6.1) and about sample compositions (Section 5.6.2).

5.6.1 Comparison of Inferred Populations

The **stochprofML** algorithm estimates the assumed parameterized single-cell distributions underlying the samples and; as described in Section 3.4, we can select the most appropriate number of cell populations using the BIC. Assume we have performed this estimation for samples from two different groups, cases and controls. One may in practice then want to know whether the inferred single-cell populations are substantially different between the two groups, e.g. in case the estimated log-means $\hat{\mu}_{\text{cases}}$ and $\hat{\mu}_{\text{controls}}$ are close to each other. A related question is whether the difference is biologically relevant.

We hence seek a method that can judge statistical significance and potentially reject the null hypothesis that two single-cell populations are the same; and at the same time allow the interpretation of similarity. Direct application of Kolmogorov-Smirnov or likelihood-ratio tests to the observed data is impossible here since the single-cell data is unobserved: We only measure the overall gene expression of pools of cells. Calculation of the Kullback-Leibler divergence of the two distributions would be possible; however, it is not target-oriented for our application where we seek an interpretable measure of similarity rather than a comparison between more than two population densities.

For our purposes, we use a simple intuitive measure of similarity — the overlap of two PDFs, that is the intersection of the areas under both PDF curves:

$$\text{OVL}(f, g) = \int_{-\infty}^{\infty} \min\{f(x), g(x)\} dx \tag{5.10}$$

for two continuous one-dimensional PDFs f and g (see also Pastore and Calcagnì, 2019). The overlap lies between zero and one, with zero indicating maximum dissimilarity and one implying (almost sure) equality. In our case, we are particularly interested in the overlap of two lognormal PDFs:

```
OVL_LN_LN <- function(mu_1, mu_2, sigma_1, sigma_2) {
 f1 <- function(x){dlnorm(x, meanlog = mu_1, sdlog = sigma_1) }
 f2 <- function(x){dlnorm(x, meanlog = mu_2, sdlog = sigma_2) }
 f3 <- function(x){pmin(f1(x), f2(x))}
 integrate(f3, lower = 0, upper = Inf, abs.tol = 0)$value
}
```

Certainly, the formula can also be applied to discrete distributions (e.g. the NB distributions), where the integral is exchanged for a sum.

Figure 5.12 shows examples of such overlaps. Here, the overlap ranges from 12% for two quite different distributions to 86% for two seemingly similar distributions. The question is where to draw a cutoff, that is, at what point we decide to label two distributions as different. Current literature considers two cases: Either the parametric case (e.g. Inman and Bradley, 1989), where both distributions are given by their distribution families and parameter values; or the non-parametric case (e.g. Pastore and Calcagnì, 2019), where observations (but no theoretical distributions) are available for the two populations. Our application builds a third case: On the one

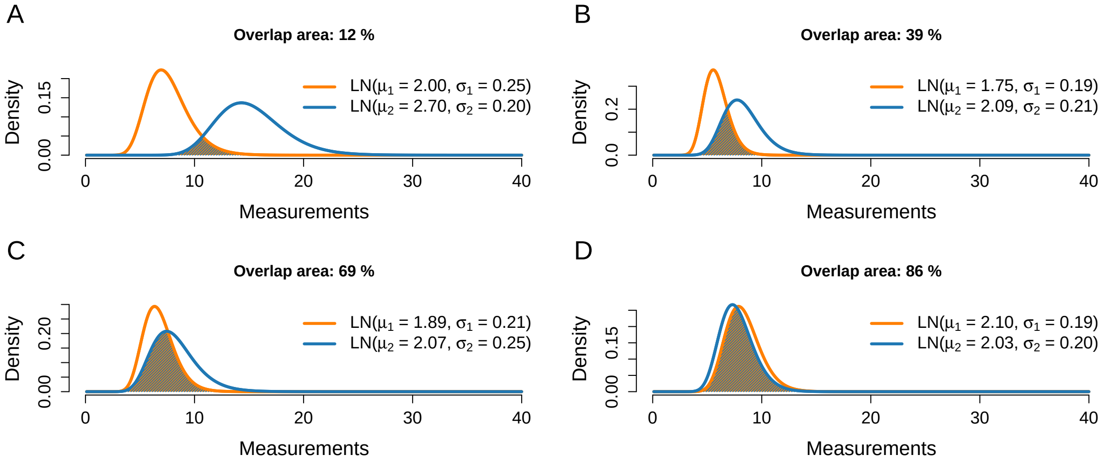


Figure 5.12: Four examples of overlapping PDFs, together with the overlap area as defined in Equation (5.10).

hand, we want to compare two parametric distributions, but the model parameters are just given as estimates based on (potentially small) datasets, thus they are uncertain; on the other hand, we do not directly observe the single-cell gene expression but just the pooled one. To address this issue, we suggest to again take into account the original data that led to the estimated parametric PDFs. As an example, assume that we consider two sets of pooled gene expression, one for a group of cases and one for a group of controls. In both groups, pooled gene expression is available as 10-cell measurements, but the two groups differ in sample size. Let's say the cases contain 50 samples and the controls 100. We assume the LN-LN model with

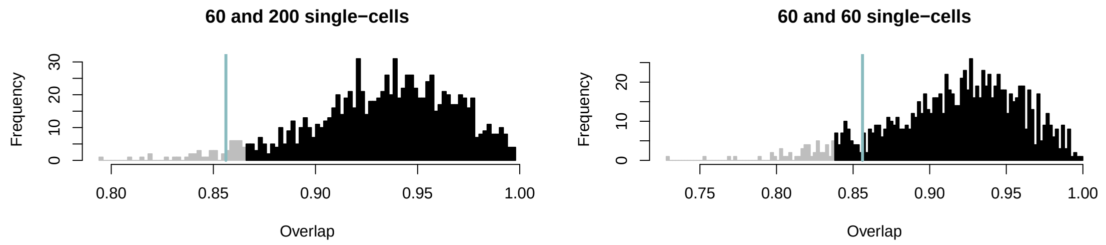


Figure 5.13: Variability of the overlap between the PDFs of the two distributions described in Figure 5.12D. The panels show histograms of $N = 1,000$ simulated overlap values which are simulated as described in the main text. *Left:* We assume that the estimates of the orange distribution relied on 60 single cells and the blue distribution on 200 single cells. *Right:* For both distributions, parameters are assumed to be estimated on 60 single cells. The 86% overlap of the original PDFs from Figure 5.12D, i. e. $\mathrm{LN}(\hat{\mu}_{1,cases} = 2.10, \hat{\sigma}^2_{cases} = 0.19^2)$ and $\mathrm{LN}(\hat{\mu}_{1,controls} = 2.03, \hat{\sigma}^2_{controls} = 0.20^2)$, is marked in turquoise. The light gray bars of the histogram indicate values below the empirical 5%-quantile. If the original overlap falls into this range, we reject the null hypothesis that both distributions identical.

two populations and estimate the mixture and population parameters using the **stochprofML** algorithm separately for each group, leading to estimates $\hat{p}_{\text{cases}}$, $\hat{\mu}_{1,\text{cases}}$, $\hat{\mu}_{2,\text{cases}}$, $\hat{\sigma}_{\text{cases}}$ and $\hat{p}_{\text{controls}}$, $\hat{\mu}_{1,\text{controls}}$, $\hat{\mu}_{2,\text{controls}}$, $\hat{\sigma}_{\text{controls}}$. We now aim to assess whether the first populations in both groups have identical characteristics, i. e. whether $\text{LN}(\hat{\mu}_{1,\text{cases}}, \hat{\sigma}^2_{\text{cases}})$ and $\text{LN}(\hat{\mu}_{1,\text{controls}}, \hat{\sigma}^2_{\text{controls}})$ are estimates of the same distribution. Figure 5.12 displays the single-cell PDFs of the first population and their overlaps for various values of the estimates. For example, in Figure 5.12D, the orange curve shows the single-cell PDF of population 1 inferred from the cases, yielding $\text{LN}(\hat{\mu}_{1,\text{cases}} = 2.10, \hat{\sigma}^2_{\text{cases}} = 0.19^2)$, and the blue one shows the inferred single-cell PDF of population 1 from the controls, $\text{LN}(\hat{\mu}_{1,\text{controls}} = 2.03, \hat{\sigma}^2_{\text{controls}} = 0.20^2)$. The overlap of these two inferred PDFs equals 86%.

We now aim to test the null hypothesis that the underlying populations $\text{LN}(\mu_{1,\text{cases}}, \sigma^2_{\text{cases}})$ and $\text{LN}(\mu_{1,\text{controls}}, \sigma^2_{\text{controls}})$ are the same versus the experimental hypothesis that they are different. We perform a sampling-based test: Taking into account the inferred population probabilities $\hat{p}_{\text{cases}}$ and $\hat{p}_{\text{controls}}$ and the number of samples and cells in the data, we can estimate the number of cells which the estimates $\hat{\boldsymbol{\theta}}_{\text{cases}}$ and $\hat{\boldsymbol{\theta}}_{\text{controls}}$ relied on. The larger this cell number, the less expected uncertainty about the estimated population distributions $\text{LN}(\hat{\mu}_{1,\text{cases}}, \hat{\sigma}^2_{\text{cases}})$ and $\text{LN}(\hat{\mu}_{1,\text{controls}}, \hat{\sigma}^2_{\text{controls}})$ (neglecting the impact of pool sizes).

In our example, let $\hat{p}_{\text{cases}} = 12\%$. Then, approximately 12% of the 500 cells from the cases group (50 × 10-cell samples) belonged to population 1, that is 60 cells. For $\hat{p}_{\text{controls}} = 20\%$, 200 cells were expected to be from the first population (that is 20% of 1,000 cells, coming from the 100 × 10-cell measurements for the controls). In our procedure, we compare parameter estimates that are based on the respective numbers of single cells, i. e. 60 cells for cases and 200 cells for controls. We perform the following steps:

- Calculate $\text{OVL}_{\text{original}}$, the overlap of the PDFs of $\text{LN}(\hat{\mu}_{1,\text{cases}} = 2.10, \hat{\sigma}^2_{\text{cases}} = 0.19^2)$ and $\text{LN}(\hat{\mu}_{1,\text{controls}} = 2.03, \hat{\sigma}^2_{\text{controls}} = 0.20^2)$.
- Under the null hypothesis, the two distributions are identical. We approximate the parameters of this identical distribution as $\tilde{\mu}_{1,\text{mean}} = (\hat{\mu}_{1,\text{cases}} + \hat{\mu}_{1,\text{controls}})/2$ and
  $\tilde{\sigma}_{\text{mean}} = (\hat{\sigma}_{\text{cases}} + \hat{\sigma}_{\text{controls}})/2$.
- Repeat $N = 1,000$ times:
  - Draw dataset A of size 60 from $\text{LN}(\tilde{\mu}_{1,\text{mean}}, \tilde{\sigma}^2_{\text{mean}})$.
  - Draw dataset B of size 200 from $\text{LN}(\tilde{\mu}_{1,\text{mean}}, \tilde{\sigma}^2_{\text{mean}})$.
  - Estimate the log-mean and log-sd for these two datasets using the method of maximum likelihood, yielding $\hat{\mu}_A$, $\hat{\sigma}_A$, $\hat{\mu}_B$ and $\hat{\sigma}_B$.
  - Calculate $\text{OVL}\left(f_{\text{LN}(\hat{\mu}_A, \hat{\sigma}^2_A)}, f_{\text{LN}(\hat{\mu}_B, \hat{\sigma}^2_B)}\right)$.
- Sort the N overlap values and select the empirical 5% quantile $\text{OVL}_{0.05}$.
- Compare the overlap from the original data to this quantile:

– If $\mathrm{OVL}_{original} \leq \mathrm{OVL}_{0.05}$, the null hypothesis that both populations are the same can be rejected.

– If $\mathrm{OVL}_{original} > \mathrm{OVL}_{0.05}$, the null hypothesis cannot be rejected.

This procedure is related to the idea of parametric bootstrap with the difference that our original data is on the n-cell level and the parametrically simulated data is on the single-cell level.

The left panel of Figure 5.13 shows one outcome of the above-described procedure (i. e. the stochastic, sampling-based algorithm was run once) with the above-specified values of the parameter estimates. Here, $\mathrm{OVL}_{original}$ lies in the critical range such that we reject the null hypothesis that the gene expression of the populations in question stem from the same lognormal distribution. We thus assume a difference here. The right panel of Figure 5.13 demonstrates the importance of taking into account the number of cells which the original estimates were based on: Here, we show one outcome of the above described steps, but this time we assume that for the control group there were only 30 10-cell samples (i.e. 300 cells in total). With the same population fraction as before ($\hat{p}_{\text{controls}} = 20\%$), the datasets B now contain only 60 cells. Here, the value $\mathrm{OVL}_{original}$ does not fall into the critical range, and therefore we would not reject the null hypothesis that the two populations of interest are the same.

When testing for heterogeneity for several genes simultaneously, multiple testing issues should be taken into account. However, genes will not in general be independent from each other.

5.6.2 Prediction of sample compositions

The **stochprofML** algorithm estimates the parameters of the mixture model, i. e. — in case of at least two populations — the probability for each cell within a pool to fall into the specific populations. It does *not* reveal the individual pool compositions. In some applications, however, exactly this information is of particular interest. Here, we present how one can infer likely population compositions of a particular cell pool. This is done in a two-step approach via conditional prediction: First, one estimates the model parameters from the observed pooled gene expression, i. e. one obtains an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. Then, one assumes that $\boldsymbol{\theta}$ equals $\hat{\boldsymbol{\theta}}$ and derives the most probable population composition via maximizing the conditional probability of a specific composition given the pooled gene expression.

A key formula here is the conditional probability of a cell composition given the measured gene expression, which we derive here. We use the following notations and assumptions:

- The overall gene expression of a cell pool is denoted by Y and assumed a continuous/discrete random variable with PDF/PMF $f_Y(y)$.

- $L = (L_1, \ldots, L_T)$ denotes the specific cell population combinations, i. e. L_i is the number of cells of population i for all $i = 1, \ldots, T$, within a pool of $L_1 + \ldots + L_T$ cells. L is a discrete random vector with PMF $P(L = \ell)$.

- $f_{Y|L=\ell}(y)$ is the conditional PDF/PMF of the overall gene expression in a cell pool whose composition is known to equal ℓ. For shorter notation, this was referred to as $f_{(\ell_1,\ell_2,\ldots,\ell_T)}(y_i|\boldsymbol{\theta})$ in Section 5.2.3.
- In turn, $P(L = \ell|Y = y)$ is the conditional PMF of the cell pool composition given the pool gene expression measurement $Y = y$.

We use Bayes' theorem to derive the latter PMF:

$$P(L=\ell|Y=y) = \frac{f_{Y|L=\ell}(y)P(L=\ell)}{f_Y(y)} = \frac{f_{Y|L=\ell}(y)P(L=\ell)}{\sum_{j\in J} f_{Y|L=j}(y)P(L=j)}, \tag{5.11}$$

where J is the set of all possible compositions of the cell pool, i. e. the set of all vectors $(j_1,\ldots,j_T)$ with $j_i \in \mathbb{N}_0$ and $j_1+\ldots+j_T = \ell_1+\ldots+\ell_T$.
The terms in Equation (5.11) depend on the population probabilities $\boldsymbol{p} = (p_1,\ldots,p_T)$ and the gene expression model (in this work: LN-LN, rLN-LN, EXP-LN or NB-NB), characterized by its respective parameters. We assume the expression model to be fixed and denote all model parameters (including $\boldsymbol{p}$) by $\boldsymbol{\theta}$. In practice, $\boldsymbol{\theta}$ is unknown, and hence we use its maximum likelihood estimates here.

Given the estimate $\hat{\boldsymbol{p}}$ of $\boldsymbol{p}$, $L = \ell = (\ell_1,\ldots,\ell_T)$ approximately follows a multinomial distribution with parameters $n = \ell_1+\ldots+\ell_T$ and $\hat{\boldsymbol{p}}$. The PMF of the cell pool composition $(\ell_1,\ldots,\ell_T)$ hence reads

$$P(L=(\ell_1,\ldots,\ell_T)) = \binom{n}{\ell_1,\ell_2,\ldots,\ell_T}\hat{p}_1^{\ell_1}\hat{p}_2^{\ell_2}\cdots\hat{p}_T^{\ell_T},$$

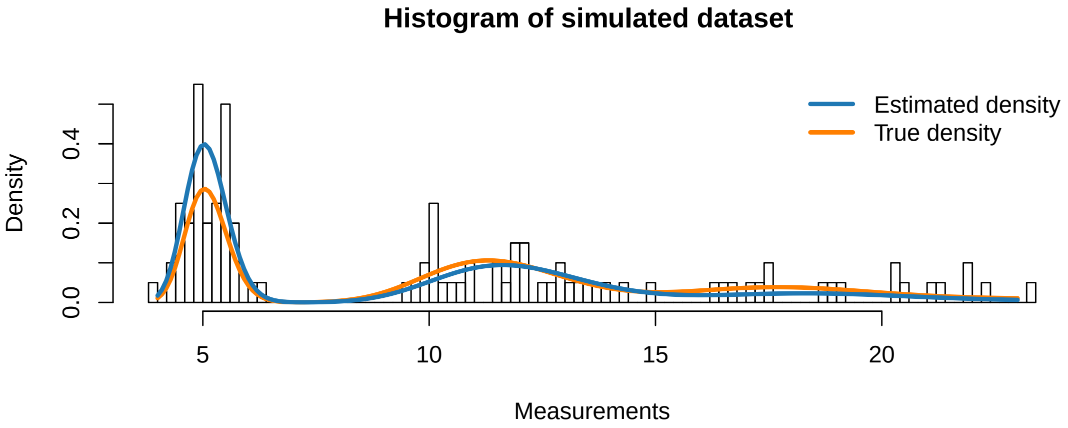


Figure 5.14: Histogram of simulated data underlying the prediction of cell pool compositions in Figure 5.15A: 100 synthetic 5-cell measurements arising from the LN-LN model with two populations with parameters $\boldsymbol{p} = (0.2, 0.8)$, $\boldsymbol{\mu} = (2, 0)$ and $\sigma = 0.2$. The PDF with true model parameters is shown in orange, the PDF with estimated parameters $\hat{\boldsymbol{p}} = (0.14, 0.86)$, $\hat{\boldsymbol{\mu}} = (2.04, 0)$ and $\hat{\sigma} = 0.20$ in blue.

where $\binom{n}{\ell_1,\ell_2,\ldots,\ell_T} = \frac{n!}{\ell_1!\,\ell_2!\cdots\ell_T!}$ is the multinomial coefficient. With this, the conditional PMF of the cell pool composition given the pooled gene expression measurement Y reads:

$$\begin{aligned} P(L=\ell|Y=y) &= \frac{f_{Y|L=\ell}(y;\hat{\boldsymbol{\theta}})\binom{n}{\ell_1,\ell_2,\ldots,\ell_T}\hat{p}_1^{\ell_1}\hat{p}_2^{\ell_2}\cdots\hat{p}_T^{\ell_T}}{f_Y(y;\hat{\boldsymbol{\theta}})} \\ &= \frac{f_{Y|L=\ell}(y;\hat{\boldsymbol{\theta}})\binom{n}{\ell_1,\ell_2,\ldots,\ell_T}\hat{p}_1^{\ell_1}\hat{p}_2^{\ell_2}\cdots\hat{p}_T^{\ell_T}}{\sum_{j\in J} f_{Y|L=j}(y;\hat{\boldsymbol{\theta}})\binom{n}{j_1,j_2,\ldots,j_T}\hat{p}_1^{j_1}\hat{p}_2^{j_2}\cdots\hat{p}_T^{j_T}}. \qquad (5.12) \end{aligned}$$

We evaluate this procedure via a simulation study. As before, we simulate data using the **stochprofML** package. In particular, we use the LN-LN model with two populations with parameters $\boldsymbol{p} = (0.2, 0.8)$, $\boldsymbol{\mu} = (2, 0)$ and $\sigma = 0.2$. Each simulated measurement shall contain the pooled expression of $n = 5$ cells, and we sample $k = 100$ such measurements. We store the original true cell pool compositions from the data simulation step in order to later compare the composition predictions to the ground truth. Having generated the synthetic data, we apply **stochprofML** to estimate the model parameters $\boldsymbol{p}$, $\boldsymbol{\mu}$ and σ. Figure 5.14 shows a histogram of one simulated data set along with the PDF of the true population mixture and the PDF of the estimated population mixture (that is the LN-LN model with parameters $\hat{\boldsymbol{p}} = (0.14, 0.86)$, $\hat{\boldsymbol{\mu}} = (2.04, 0)$ and $\hat{\sigma} = 0.20$).

Next, we calculate the conditional PMF (see Equation (5.12)) for each possible population composition conditioned on the particular pooled gene expression measurement. Figure 5.15A and Table 5.6 show results for the first six (out of 100) pooled measurements.

In particular, Figure 5.15A displays the conditional PMF of all possible compositions (i.e. k times population 1 and $5-k$ times population 2 for $k \in \{0, 1, \ldots, 5\}$). Blue bars stand for these probabilities when $\hat{\boldsymbol{\theta}}$ is used as model parameter value. Orange stands for the hypothetical case where the true value $\boldsymbol{\theta}$ is known and used. These two scenarios are in good agreement with each other.

We regard the most likely sample composition to be the one that maximizes the conditional PMF (maximum likelihood principle). The true composition (ground truth) is marked with a black box around the blue and orange bars. We observe in Figure 5.15A that the composition is in all six cases inferred correctly and mostly unambiguously. Only for the fifth measurement, there is visible probability mass on a composition other than the true one. In fact, it is the only pool (out of the six considered ones) with two cells from the first population. Alternatively to the maximum likelihood estimator, one can also regard the expected composition — the empirical weighted mean of numbers of cells in the first population — or confidence intervals for this number. The respective estimates for the first six measurements of the dataset are shown in Table 5.6. The results are consistent with the interpretation of Figure 5.15A.

Certainly, the precision of the prediction depends on the employed pool sizes, the underlying true model parameters and how reliably these were inferred during the first step. We showed in Section 5.4 that larger cell pools lead to less precise parameter inference. Hence, we repeat the prediction of sample compositions on another dataset,

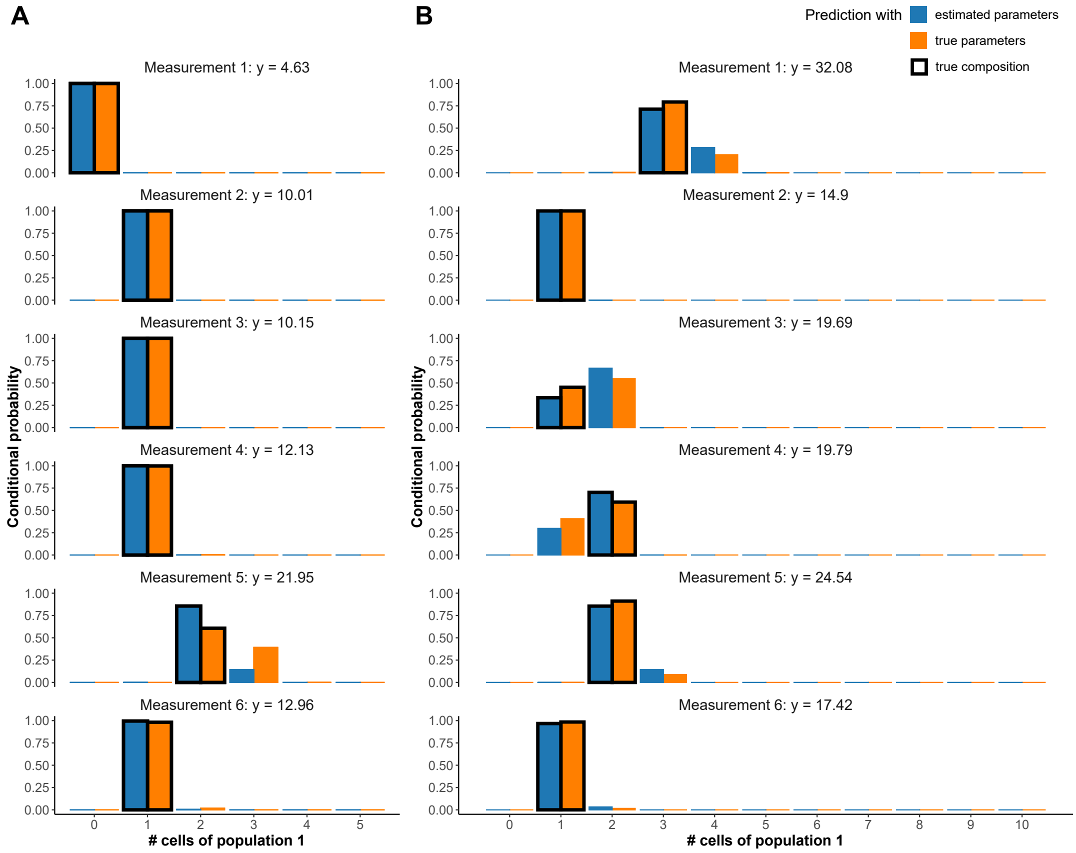


Figure 5.15: Estimation of cell pool compositions in the two-population LN-LN model: Conditional probabilities of numbers of cells from the first population in the first six measurements of the synthetic datasets described in the main text and in Figure 5.14, given the respective pooled gene expression measurement. Blue bars show the conditional probabilities using estimated model parameters, and orange bars show those when using the true parameters. True cell numbers from the first population are marked with a black box around the bars. Results for (A) simulated 5-cell data and, (B) 10-cell data.

this time based on 10-cell pools. All other parameters remain unchanged. The resulting conditional probabilities are depicted in Figure 5.15B. Since $p = 0.2$, one expects on average two cells to be from the first population in each 10-cell pool. As in the previous 5-cell case, most predictions show a clear pattern. However, probability masses are spread more widely. Measurements 3 and 4 exemplify that almost identical gene expression measurements ($y = 19.69$ and $y = 19.79$) can arise from different underlying pool compositions (two times population 1 in measurement 3 vs. three times population 1 in measurement 4). For more similar population parameters, the estimation will get worse, which will then propagate to the well composition prediction. In such cases, to predict the pool compositions, one may use additional parallel measurements of other genes that might separate the population better by their different expression profiles while the pool composition stays the same across genes.

Estimator for # of cells in pop. 1		**Measurement index**						**# of hits**
		1	**2**	**3**	**4**	**5**	**6**	
Estimated parameters	Mean	0.00	1.00	1.00	1.00	2.14	1.01	98
	MLE (CI)	0 (0,0)	1 (1,1)	1 (1,1)	1 (1,1)	2 (2,3)	1 (1,1)	98 (100)
True parameters	Mean	0.00	1.00	1.00	1.00	2.39	1.02	97
	MLE (CI)	0 (0,0)	1 (1,1)	1 (1,1)	1 (1,1)	2 (2,3)	1 (1,1)	97 (100)
True # of cells from population 1		0	1	1	1	2	1	

Table 5.6: Estimates of numbers of cells from the first population in the simulated 5-cell data described in Figures 5.14 and 5.15A and in the main text. *Columns:* Estimation results for the first six measurements from the datasets and (last column) summary across all 100 samples. *Rows:* Estimation of cell numbers are based on conditional probabilities that use either the estimated model parameters (rows 1 and 2, corresponding to blue bars in Figure 5.15A) or the true values (rows 3 and 4, orange bars). Within each of these two choices one can consider the mean number of cells from population 1 as determined by the conditional probabilities (rows 1 and 3) or the MLE that maximizes the conditional probabilities (rows 2 and 4, first value) including a 95% confidence interval that covers at least 95% of the conditional probability mass (rows 2 and 4, in parentheses). The last row shows the true pool composition. The last column shows for each estimator how many of the 100 cell numbers were inferred correctly (defined as follows: rounded mean is exact match; MLE is exact match; CI includes correct number).

5.7 Discussion and Conclusion

With the **stochprofML** package, we provide an environment to profile gene expression measurements obtained from small pools of cells. Experimentalists may choose this approach if single-cell measurements are impossible in their lab, e. g. if the drop-out rate of the tissue of interest is too high in single-cell libraries, if budget or time are limited, or if one prefers to avoid the stress which is put on the cells during cell separation. One of the latest implementations even allows to combine information from different pool sizes, in particular, to simultaneously analyze single-cell and n-cell data. Another major addition to the software now includes the discrete NB distribution, which adapted the underlying model assumptions to the discrete nature of sequencing data.

We demonstrated the usage and performance of the **stochprofML** algorithm in various examples and simulation studies. These have been performed in an idealized in silico environment. This should be kept in mind when incorporating the results into experimental planning and analysis. Subsequent interpretation of heterogeneity will be informative if based on a good model estimate. The assumption of independent expression across genes within the same tissue sample is a simplification of nature that leads to less complex parameter estimation. Previous experimental validation (Bajikar et al., 2014) provided evidence that transcriptional heterogeneity can be parameterized through stochastic profiling even for non-ideal settings such as small sample sizes or in the presence of gene-gene correlation. If populations are similar or diffuse, they may not be identified as distinct populations through **stochprofML**.

The same, however, applies to other statistical methods and also to the analysis of single-cell data. For the latter, noise is expected to be more pronounced than in n-cell pools, which again motivates the use of our method.

The optimal pool size with respect to bias and variance of the corresponding parameter estimators will depend on unknown properties such as numbers of populations and their characteristics, and also on the relationship between the pool size and the amount of technical measurement noise. The latter aspect has been excluded from the studies here but further supports the application of stochastic profiling. We compare the discrete **stochprofML** variant with a newly implemented Bayesian version using Stan. We show that both deliver comparable results with very much reduced runtimes in the Bayesian version. This underlines a strength of these methods for accelerating runtimes by bypassing the computationally intensive likelihoods. Other Bayesian methods such as variational Bayesian inference (Blei et al., 2017) could reduce them even further.

6 Application to Real-World Small Pool Data

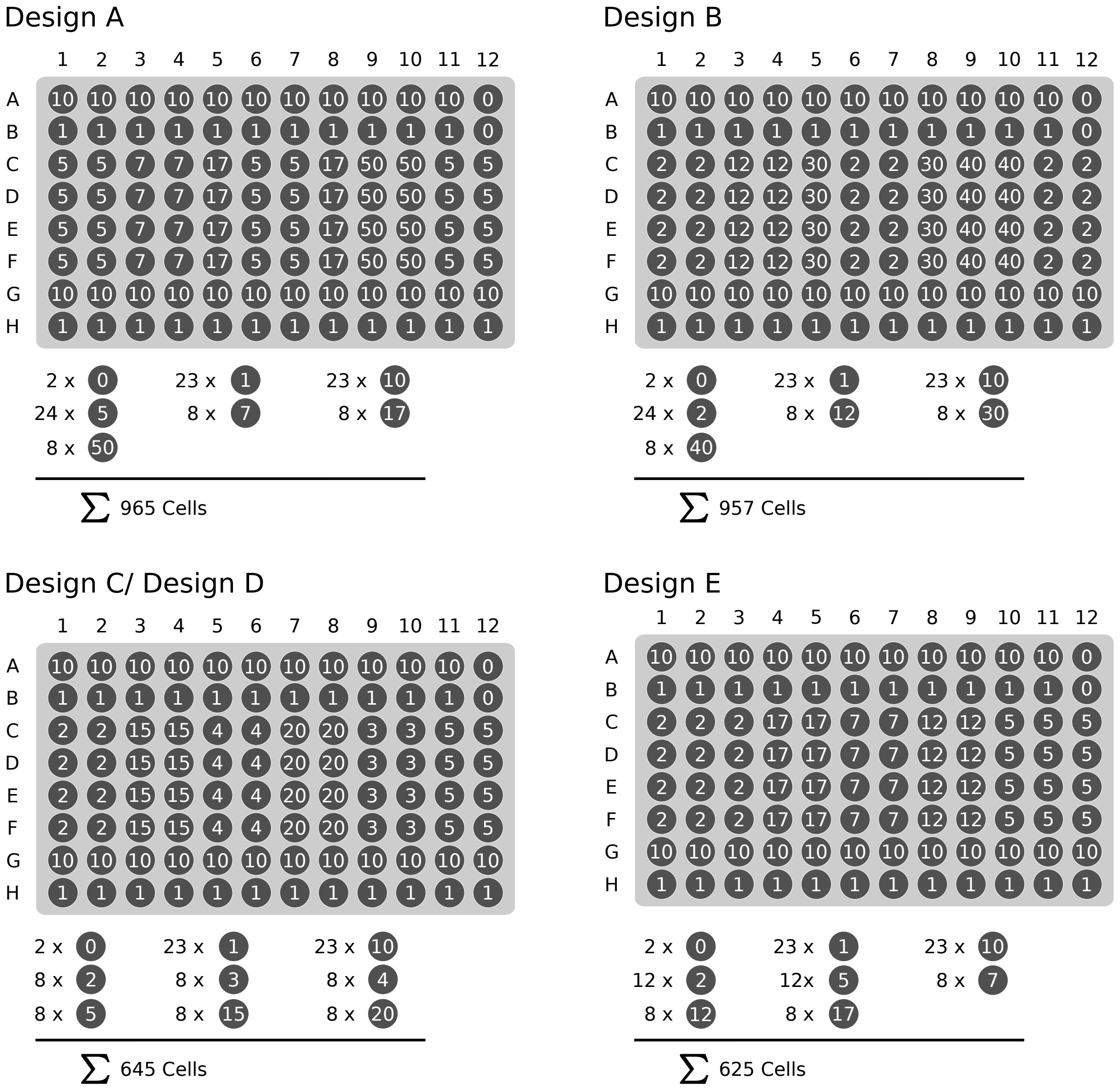
We discussed in the previous chapters discrete probability distributions that model mRNA counts and how to deconvolve single-cell population profiles from pooled cell measurements. Based on our models, in this chapter we want to explore the hypotheses that cell pooling might be advantageous to measure less noise and to better detect heterogeneities by using real data from small cell pools generated for this purpose. Therefore, we need to establish a suitable study design which is explained below.

6.1 Experimental Setup

The biological truth about inherent heterogeneity in a sample is generally unknown. Therefore we have to rely on assumptions when selecting a suitable tissue for this experiment. To investigate the detection of heterogeneity we want to analyze some heterogeneous as well as homogeneous cells in two parallel experiments. We have chosen mouse embryonic stem cells (mESC) and AML cells: The mESC were selected because they are presumably homogeneous, cheap and convenient to use (according to our collaboration partners). The AML cells were selected because we are part of the Collaborative Research Center (CRC) 1243 *Cancer Evolution: Genetic and Epigenetic Evolution of Hematopoietic Neoplasms* in Munich, where the majority of the participating cooperation partners are working on different types of leukemia. Since AML often contains several subpopulations – so called subclones (see Chapter 2.3) –, it can be assumed that this is an appropriate choice to study. Additionally, it is worth mentioning that there exist not many gene expression datasets of AML cells since this measurement is very challenging. We know from personal correspondence with our collaboration partners of the Enard lab (Johannes Bagnoli, Faculty of Biology, LMU Munich), that they contain much fewer RNA than for example mESCs which means that you simply start with less input material for the reactions. In addition, they also have more RNases, i.e. internally expressed enzymes that attack RNA. This leads to the fact that when the cells are lysed these enzymes attack the RNA at the same time and you have even less input material. Therefore one has to make sure that these RNases cannot work too well before the reverse transcription. In addition, AML cells are incredibly sensitive when it comes to thawing. Many cells do not survive this and depend on how the cells are sorted (e. g. with which stainings). Therefore, many cells are either already completely dead or at least have started to die, which also leads to the fact that the RNA is broken or even actively destroyed.

To investigate the measured mRNA material in terms of the contained cell numbers of the cell pools, we have to determine pool sizes as well as their frequencies in the experiment. In general, sequencing plates contain a finite number of wells. In order to get many samples, several plates need to be used (their number is restricted by financial constraints). Using several plates will almost surely introduce batch effects. Therefore we need to generate plate designs that include measurements of some pool sizes on each plate to ensure comparability between all plates. Figure 6.1 shows the frequencies of pool sizes on each 96 well plate. We created five different designs where each contains the same number of single cells and 10 cells with which we hope to correct possible batch effects later. Additionally, each plate contains two wells with zero cells to control the background noise. The cell numbers of the remaining wells were selected such that pool sizes up to 50 cells are measured while simultaneously the designs used in one batch contain a comparably same number of total cells. This is important when determining the sequencing depth which should be approximately equal per cell. Sequencing depth was set to 300 million reads for plates of design A and B and to 200 million reads for plates of design C, D and E.

The experiments are run in three batches. Batch 1 contains three plates of plate C, batch 2 consists of three plates of design A and of three plates of design B. Finally, batch 3 consists of five plates of design D and of five plates of design E. Taken together, 19 plates with mESC cells and another 19 plates with AML cells were sequenced. Table 6.1 shows all planned plates and pool sizes. In total, 437 single-cell measurements and the same number of 10-cell measurements were planned for each of the tissues. To be sure that the correct number of cells is contained in each well, the cells are FACS sorted into the wells. Due to a FACS error during batch 3 of the AML experiment, single-cells were sorted into the forty wells originally planned for 15-cell measurements. Therefore we miss the 15-cell measurements on these plates and got additional single-cell measurements. By this the mESC and AML experiments are no longer completely parallel but since we analyze the tissues separately and do not want to compare them one-by-one this will pose no problem.



Design A

	1	2	3	4	5	6	7	8	9	10	11	12
A	10	10	10	10	10	10	10	10	10	10	10	0
B	1	1	1	1	1	1	1	1	1	1	1	0
C	5	5	7	7	17	5	5	17	50	50	5	5
D	5	5	7	7	17	5	5	17	50	50	5	5
E	5	5	7	7	17	5	5	17	50	50	5	5
F	5	5	7	7	17	5	5	17	50	50	5	5
G	10	10	10	10	10	10	10	10	10	10	10	10
H	1	1	1	1	1	1	1	1	1	1	1	1

2 x 0 23 x 1 23 x 10
24 x 5 8 x 7 8 x 17
8 x 50

∑ 965 Cells

Design B

	1	2	3	4	5	6	7	8	9	10	11	12
A	10	10	10	10	10	10	10	10	10	10	10	0
B	1	1	1	1	1	1	1	1	1	1	1	0
C	2	2	12	12	30	2	2	30	40	40	2	2
D	2	2	12	12	30	2	2	30	40	40	2	2
E	2	2	12	12	30	2	2	30	40	40	2	2
F	2	2	12	12	30	2	2	30	40	40	2	2
G	10	10	10	10	10	10	10	10	10	10	10	10
H	1	1	1	1	1	1	1	1	1	1	1	1

2 x 0 23 x 1 23 x 10
24 x 2 8 x 12 8 x 30
8 x 40

∑ 957 Cells

Design C/ Design D

	1	2	3	4	5	6	7	8	9	10	11	12
A	10	10	10	10	10	10	10	10	10	10	10	0
B	1	1	1	1	1	1	1	1	1	1	1	0
C	2	2	15	15	4	4	20	20	3	3	5	5
D	2	2	15	15	4	4	20	20	3	3	5	5
E	2	2	15	15	4	4	20	20	3	3	5	5
F	2	2	15	15	4	4	20	20	3	3	5	5
G	10	10	10	10	10	10	10	10	10	10	10	10
H	1	1	1	1	1	1	1	1	1	1	1	1

2 x 0 23 x 1 23 x 10
8 x 2 8 x 3 8 x 4
8 x 5 8 x 15 8 x 20

∑ 645 Cells

Design E

	1	2	3	4	5	6	7	8	9	10	11	12
A	10	10	10	10	10	10	10	10	10	10	10	0
B	1	1	1	1	1	1	1	1	1	1	1	0
C	2	2	2	17	17	7	7	12	12	5	5	5
D	2	2	2	17	17	7	7	12	12	5	5	5
E	2	2	2	17	17	7	7	12	12	5	5	5
F	2	2	2	17	17	7	7	12	12	5	5	5
G	10	10	10	10	10	10	10	10	10	10	10	10
H	1	1	1	1	1	1	1	1	1	1	1	1

2 x 0 23 x 1 23 x 10
12 x 2 12x 5 8 x 7
8 x 12 8 x 17

∑ 625 Cells

Figure 6.1: Overview of the five plate designs of the mESC and AML small pool measurements. 96 well plates are used. Numbers in wells show the number of cells in this measurements. Design C is used in the first experiment run, designs A and B in the second one. The third round of experiments uses designs D and E. Note, that plate designs C and D are the same, but differ slightly in the library preparation during the experiments, see main text for more information. Well positions are named with letters A - G for the rows and with numbers 1 - 12 for the columns. Total numbers of cells measured on each plate differ.

Differences between batches do not only lie in different time points when the experiments were conducted, but also the FACS machine and the lysis buffer that was used. All three batches used the KAPA polymerase. In summer and winter 2017, when batch 1 and 2 of both cell types were generated, a SH800 sorter (Sony Biotechnology, 100 μ m chip) with "Single Cell (3 Drops)“ purity setting (see Bagnoli et al., 2018) was used to get the planned cell number in the wells. The library uses a PPP (short for Primer, ProteinaseK and Phusion Buffer) lysis buffer. In contrast to that the

	0	1	2	3	4	5	7	10	12	15	17	20	30	40	50
3x A	6	69	0	0	0	72	24	69	0	0	24	0	0	0	24
3x B	6	69	72	0	0	0	0	69	24	0	0	0	24	24	0
3x C	6	69	24	24	24	24	0	69	0	24	0	24	0	0	0
5x D	10	*115*	40	40	40	40	0	115	0	*40*	0	40	0	0	0
5x E	10	115	60	0	0	60	40	115	40	0	40	0	0	0	0
Total	38	437	196	64	64	196	64	437	64	64	64	64	24	24	24

Table 6.1: Overview of all sequenced pool sizes. This was the planned setup, for both AML and mESC experiments. In practice during preparation of the five AML plates of Design D, some error occurred, and instead of 40 15-cell measurements, additional 40 single-cells were sequenced, see numbers in italics. This results into 155 single cells and no 15-cell measurements.

cells of batch 3 were sorted in winter 2019 using a BD Aria III FACS machine and lysis buffer was changed to PPi (Primer, Phusion Buffer and RNAse inhibitor).
The experiment described above was performed in cooperation with labs of the CRC. The AML cells were derived from PDX mouse models (see Section 2.4) generated by the Jeremias lab (Research Unit Apoptosis in Hematopoietic Stem Cells, HMGU and Dr. von Hauner Children's Hospital, LMU Munich). Homogeneous mESC cells for comparisons were provided by the Leonhardt lab (Faculty of Biology, LMU Munich). Library preparation and sequencing using mcSCRB-seq (see Section 2.2.4) was performed by the Enard lab (Faculty of Biology, LMU Munich).

6.2 Experiment Output and Further Data Processing

After sequencing, data pre-processing has to be performed. We use the zUMIs pipeline (see Section 2.2.6) to map the sequences to the wells and to the reference genome. The sequencing depth only defines the (planned) number of raw reads of all material on one plate. The raw reads include reads that cannot be mapped to a well or the reference genome.
If we look at the total reads for both cell types we see a positive dependency on the pool size (Figure 6.2, left). This dependency seems to be different for the three batches. In contrast, the UMI content does not show a systematic difference for the batches. In detail, when normalizing the total UMIs to the contained cell number of each measurement, these normalized UMI counts are fairly constant by clustering around a horizontal line which is the same for the three batches (Figure 6.2B). However, the single-cells display a greater variance, especially in batch 3 with many larger measurements.

6.2.1 Downsampling

To make sure that each well was comparably often sequenced – which might cause the effect that the single-cells sometimes show higher UMI numbers – we downsample

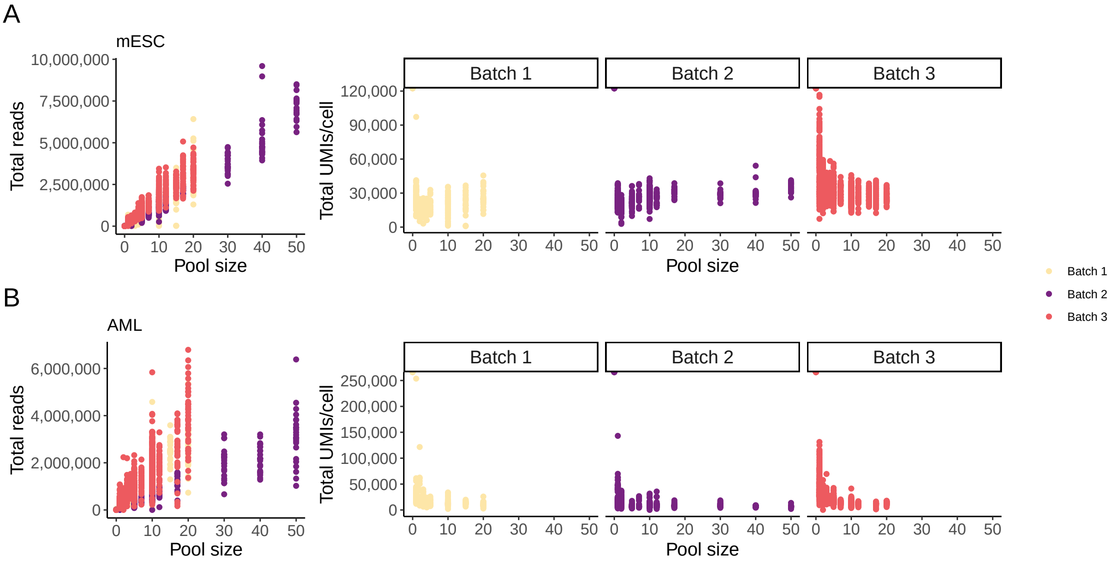


Figure 6.2: Total reads (left) and total UMIs per cell number (right) versus the number of cells contained in each sample separated by batches. (A) shows the mESC data and (B) the AML data. Colors identify the three experimental batches.

the raw reads per cell for each well to 76,877 raw reads per cell in the mESC dataset and 73,008 raw reads per cell in the AML dataset. These numbers were selected in an early analysis step of the raw reads in the first two batches. The goal was to find a high raw read number to which we can downsample each cell without loosing many measurements that contain less raw reads per cell. However, during the following downsampling we loose some measurements (32 in the mESC dataset and 27 in the AML dataset). After this, each measurement has exactly the same raw read number per cell. As before, we are interested in the mapped reads, i. e. sequences that can be assigned to genes in the reference genome and their collapse to UMI counts. Figures 6.3 depicts the total reads and UMI counts per cell per sample after these downsamplings for both cell types. Compared to Figure 6.2 the counts reduced to approximately a fifth in reads and a third in UMIs per cell. Hence, much information that was originally contained in the data is ignored when using the downsampled data. However, after downsampling the batches are depicted more clearly in the read counts, which were previously more blurred. The single-cell outliers in the UMIs are still there and could not be removed by downsampling.

6.2.2 Batch Effects of Merged Datasets

The next step is to identify possible batch effects. Since the three batches stand for three experiments and the results are given in three different data files, they need to be merged. Not all genes are present in all three batches which poses a problem in batch correction and gene analysis. A gene that is completely missing in one batch cannot be corrected or added in the dataset where it is completely missing and thus can this gene be used to identify the batches. This is why we choose to only keep

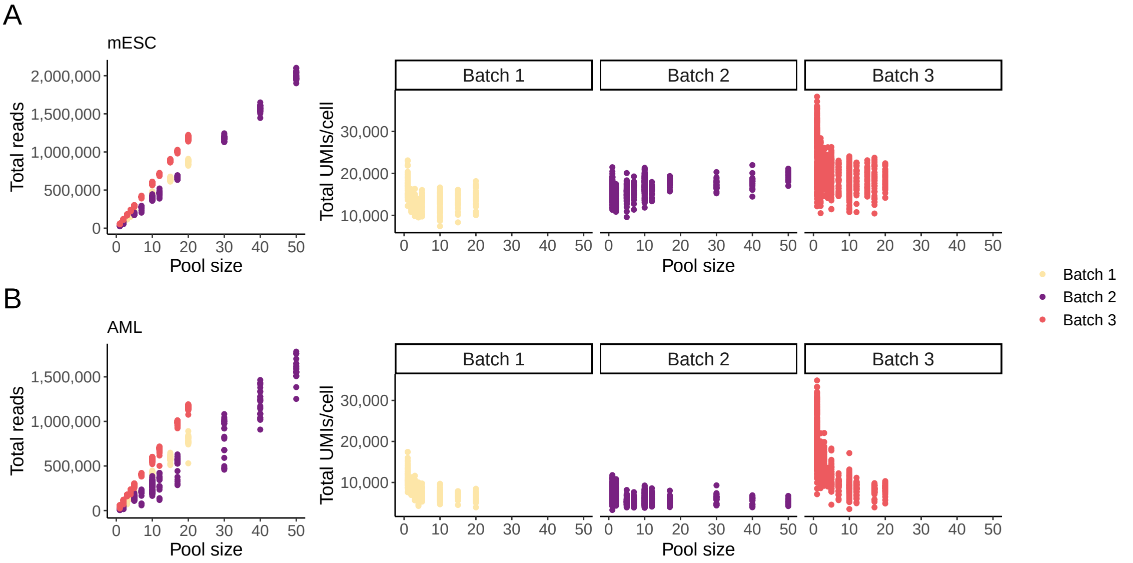


Figure 6.3: Total reads (left) and total UMIs per cell number (right) after downsampling versus the number of cells contained in each sample separated by batches. (A) shows the mESC data and (B) the AML data. Colors identify the three experimental batches.

the genes present in all batches. Hence, after merging, the complete mESC dataset consists of 22,331 genes and 1,753 observations and the AML dataset contains 27,838 genes in 1,741 observations. The UMAPs (McInnes et al., 2020) in Figures 6.4 and 6.5 visualize possible contained batch effects. We can identify clearly the batch effects introduced by the three experiment runs. Additionally the cells cluster by growing pool sizes. Since we want to analyze the effect of pool sizes, we do not want to correct for them. Therefore a batch effect correction is conducted by using the tool ComBat-seq (Zhang et al., 2020) – the new version of ComBat (Johnson et al., 2006) tailored to sequencing data – in which we enter the three batches for correction. The lower plots of Figures 6.4 and 6.5 show the UMAP of the downsampled data corrected in this way. We can see a huge improvement of mixing of the batches on the right, but a preservation of the cell number ordering. The same can be done to the original (not downsampled dataset). The corresponding figures can be found in Appendix I in Figures I.1 and I.2.

6.3 Noise Reduction by Cell Pooling

One of the aims of this study was to investigate if we can reduce technical noise by cell pooling. Since in larger pools more mRNA material is available (percentage wise) less material is lost for example during library preparation. Figure 6.2 already showed that the larger pool sizes led to more counts, but now we want to investigate this further.
Our first step is to model the relationship between total UMI counts of a well and its cell number. Especially in Chapter 4 we have put emphasis on using discrete

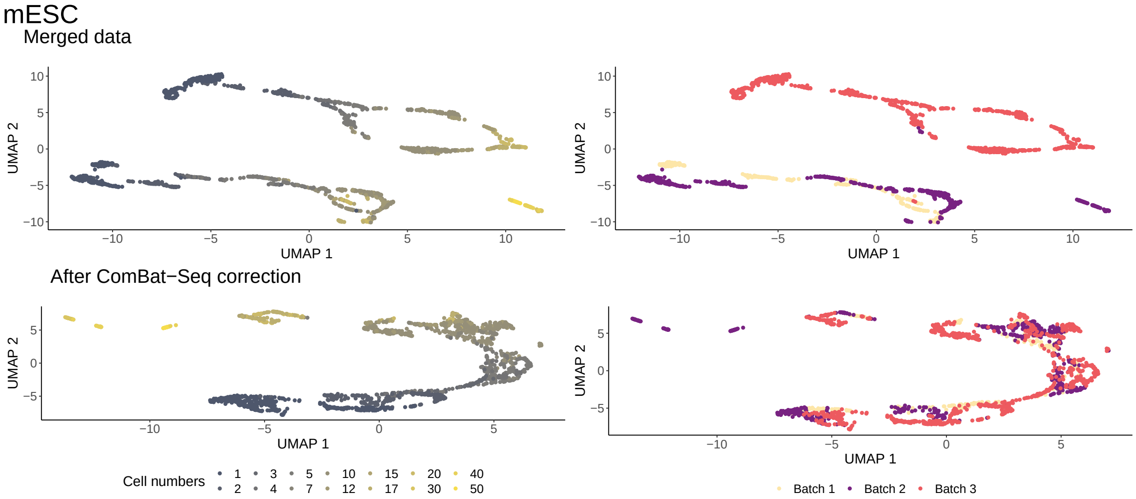


Figure 6.4: UMAPs of the UMIs of the merged downsampled mESC datasets before (top row) and after batch correction via ComBat-seq (bottom row). Colors identify cell numbers (left) and the three experimental batches (right).

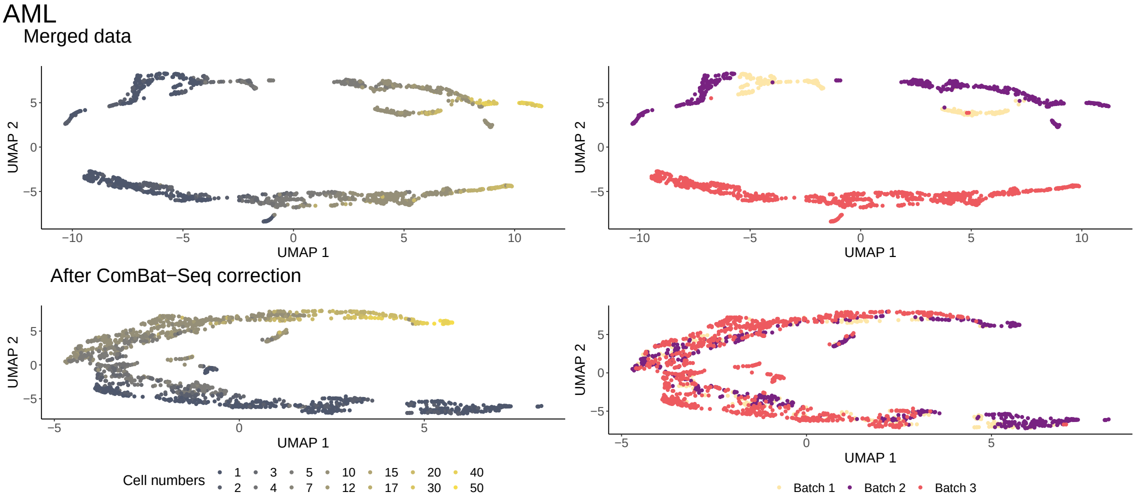


Figure 6.5: UMAPs of the UMIs of the merged downsampled AML datasets before (top row) and after batch correction via ComBat-seq (bottom row). Colors identify cell numbers (left) and the three experimental batches (right).

distributions to model discrete data. Since UMI counts are discrete, we choose to model them analogously by using the flexible NB distribution (see Definition A.8). We use a GAMLSS NB regression model (Stasinopoulos et al., 2017) to model the ComBat-seq corrected dataset with cell numbers as covariates. Figure 6.6 shows these results for both cell types. Note that we do not use the downsampled dataset here since we are interested in all the information we could get out of the data.

From previous studies (Amrhein and Fuchs, 2020b, Bajikar et al., 2014), we know that 10 cells are a reasonable pool size to use stochastic profiling on. Therefore, we include a linear relationship induced by the mean UMI counts of the 10-cell measurements

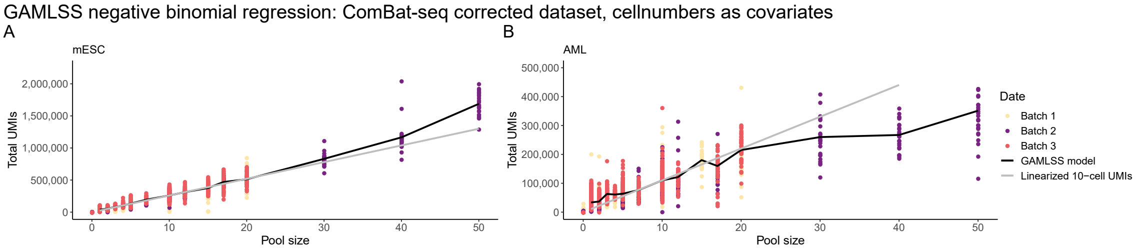


Figure 6.6: GAMLSS NB regression model using the ComBat-seq corrected UMI datasets of the mESC data (A) and the AML data (B). Cell numbers serve as covariates. For comparison the induced linear relationship by the 10-cell UMI content is added in gray.

to compare this to the model fitting of the GAMLSS. The mESC data clearly shows that the UMI content can be modeled by a roughly linear dependency, similar to the added 10-cell linearization. Note that single-cells but especially the UMI counts of the large pool sizes lie above this straight line and therefore contain slightly more material per cell than the 10-cell samples would indicate. This is different in the AML dataset. There it would be a great simplification to call the relationship of UMI counts and cell numbers linear. Small pool sizes measure more UMIs per cell while large pool sizes measure much less than expected. Similar results can be seen on an alternative approach using the GAMLSS negative binomial (NB) regression on the non corrected dataset that distinguishes between the batches. Again the cell numbers serve as covariates but additionally the batches are included as mixed effects. The plots can be found in Appendix I in Figure I.3.

Next, we want to look at this at the gene level. Pooling cells can only show its advantages with weakly expressed genes. These genes show often no expression or only noise in single-cells but might have a real signal in larger cell pools. For this reason we look at low expressed genes in the following.

We will only look at single-cells, 2-cell pools, 5-cell pools and 10-cell pools. In Figure 6.7 we zoomed into the normalized mean UMI counts of the different pool sizes in contrast to the single-cell means for all genes of the dataset. The complete figure can be found in Appendix I in Figure I.4. In a perfect world we would expect the points to lie on the gray line which describes the case that both means are the same. The mESC data is close to this case which confirms what we saw in the (linear) result of the GAMLSS model. In contrast, the AML data shows that the mean UMI counts of single-cell measurements are often higher than the normalized means of the pools. This is not what we expected, since our hypothesis was that pooling reduces measurement errors and results in higher counts but confirms what we have seen in the GAMLSS model fits where the single-cells have a higher UMI expression than expected. Here we see that this does not only hold on the total UMI level but also on the gene level.

Taken together, we cannot support the assumption that larger cell pools reduce noise and therefore we get higher UMI counts per cell based on the data generated in this study. Still we can confirm that cell pools consisting of more cells return higher UMIs and therefore contain more information than single-cells. In the homogeneous

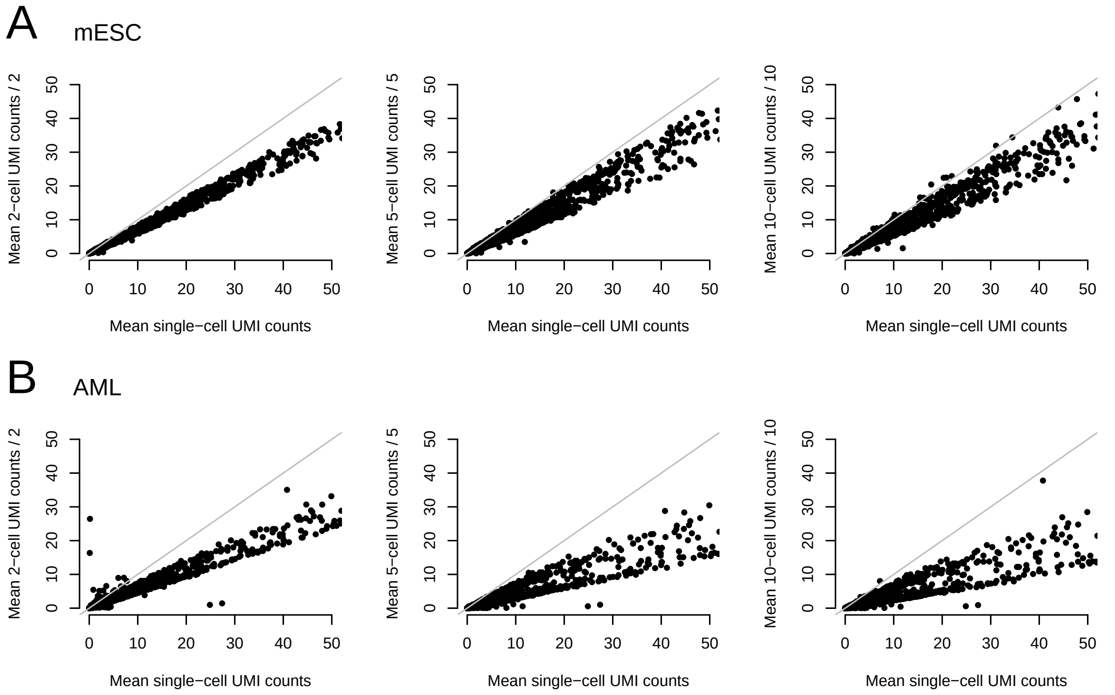


Figure 6.7: Mean single-cell UMI counts per gene compared to the normalized mean UMI-count per cell for 2-cell pools, 5-cell pools and 10-cell pools of the mESC dataset (A) and the AML dataset (B). The gray line describes the case that both means are the same. Here we zoomed in to focus on weakly expressed genes. The complete figure can be found in Appendix I in Figure I.4.

mESC dataset we can see a roughly linear relationship of UMI counts and pool sizes. The AML data is more challenging and higher cell pools seem to show some kind of saturation so that information is not growing linearly with cell sizes. We already described above that it is more challenging to measure AML cells than mESC cells. In detail, ESCs or other cells can be kept in culture (this is very difficult with AML cells, which is why there are PDX models, see Section 2.4). This is why you have to take these frozen cells, which are not very healthy in contrast to very healthy mESCs directly from the cell culture. It might be that not all subclones in the AML cells could not be measured as single-cells or that they carried very little information so that they got filtered out. Maybe these are contained in the larger pool sizes and therefore their content does not increase linearly compared to the small pool sizes. Also note that the mcSCRB-seq protocol was adapted and improved to work especially well for single-cell measurements and does not generate many dropouts. Therefore the single-cell measurements here are already pretty good and do not carry much noise at all.

6.4 Heterogeneity Detection by Cell Pooling

Another aim of this study is to investigate if heterogeneities can be detected in UMI measurements of small cell pools. Since we saw before that the measurement information is not necessarily linearly increasing with cell numbers this is quite challenging. Moreover, this violates one of the main assumptions when using our **stochprofML** algorithm. Nevertheless, we saw that in larger cell pools more UMI counts could be measured than in smaller cell pools. For this reason we will not use different pool sizes in one analysis. We use the NB-NB model of our **stochprofML** algorithm on the single-cells, the 2-cell pools, the 5-cell pools and the 10-cell pools separately. Next, we compare the results of the inferred heterogeneity in the different pool sizes. From Table 6.1, we know that we planned to have 437 single-cells and 10-cell measurements and 196 2-cell and 5-cell measurements. In practice, we have 430 single-cell, 186 2-cell, 196 5-cell and 427 10-cell observations in the mESC data set and 471 single-cell, 188 2-cell, 186 5-cell and 429 10-cell observations in the AML data set. We select 50 genes that are likely to be bimodal in the single-cell data of both datasets by looking at possible bimodality in the kernel estimates of the single-cell data. Then we apply the NB-NB model of the **stochprofML** algorithm (see Chapter 5 for details) to estimate model parameters for one or two populations. With this we get eight fits per gene. For each pool size we use model selection via BIC, given in Formula (3.9) to decide if the sample is homogeneous or rather heterogeneous with two populations. Our hypothesis is that the samples of the mESC dataset select more often the one population model and are thus homogeneous, while the AML genes show certain heterogeneity by selecting more often the two-population model. Additionally, we are interested if possible heterogeneities can be better detected in larger pools (e. g. i10 cells) than in single-cell data or small pools. The collapsed results are depicted in Table 6.2. More than half (26 genes) of the selected mESC genes seem to be homogeneous in all four pool sizes. Another 14 genes show homogeneity in all but one of the four subsets of a gene. Surprisingly most of them inferred heterogeneity in the small pools, i. e. in the single-cells or the 2-cell pools. 7 genes show heterogeneity in 3 subsets and also here mostly in the single-cell dataset. This confirms our expectation that the mESC data is thought to be a homogeneous dataset. Since, we selected the 50 genes based on the single-cell profiles we find there heterogeneity in these subsets if any.

The AML data unexpectedly shows an interesting picture: Although more heterogeneity is discovered overall, most of the genes (33 genes) show no heterogeneity or only in one or two subsets of the genes. If heterogeneity is found it is present in the subsets where the pool size is rather small. We could only find in 4 genes heterogeneity in three subsets and overall only in 6 genes the heterogeneity is found in the larger pools that contain 5 or 10 cells. Hence, compared to the 50 mESC genes we could infer less homogeneity.

Since this selection of the 50 genes was rather random, based on the single-cell profiles and not based on any biological knowledge, we additional selected 34 interesting AML genes (see, Boyd et al., 2018, Herold et al., 2018, Ng et al., 2016) and repeated

		combinations of selected populations															
sub datastets	single-cell	1	2	1	1	1	2	2	2	1	1	1	1	2	2	2	2
	2-cell	1	1	2	1	1	2	1	1	2	2	1	2	1	2	2	2
	5-cell	1	1	1	2	1	1	2	1	2	1	2	2	2	1	2	2
	10-cell	1	1	1	1	2	1	1	2	1	2	2	2	2	2	1	2
# genes	mESC (50 genes)	26	9	1	2	2	3	0	4	2	0	0	0	0	1	0	0
	AML (50 genes)	12	17	3	0	1	12	1	0	0	0	0	0	1	1	2	0
	AML II (34 genes)	31	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table 6.2: Collapsed result of model selection via BIC of stochprofML NB-NB model fits of 50 mESC, 50 AML genes that are presumably heterogeneous. A second set of 34 interesting AML genes was selected via literature. The observations were grouped into four sub datasets which only contain single-cells, 2-cell pools, 5-cell pools and 10-cell pools measurements, respectively. The NB-NB model of our **stochprofML** algorithm was applied for one and two populations on each sub-dataset. The number of genes describes how often the combination of specific population was chosen for the four sub-datasets.

the population inference. However, we could hardly infer any heterogeneity in these genes. Nearly all of them (31 genes) infer one population in all four sub-datasets. Only 3 genes found heterogeneity in the single-cell data, one of which also showed heterogeneity in the 10-cell sub dataset.

6.5 Discussion and Conclusion

With this real-world study on sequenced small pools of homogeneous and heterogeneous data we wanted to investigate if and how measurement noise decreases with pool sizes and if we can detect heterogeneities in such small pool data. We can confirm that gene expression information such as mRNA counts via reads or UMIs increase with additional cell numbers in the measured pools. The slope and details on this increase is dependent on the used tissue and its heterogeneity but probably also on the sequencing protocol. Batch effects might influence this additionally. These will be always present since to our knowledge sequencing cell pools can only be performed on plate based technologies. The use of a protocol – such as mcSCRB-seq –, that is specialized in single-cell experiments and in the extraction of the contents of sensitive tissue (like AML), leads to relatively better measurements of single-cells with rare dropout and noise, so that the advantages of cell pooling can not stick out. In general, single-cell measurements that are tailored to the specific tissue and heterogeneity are preferred if possible. For these reasons one would hardly use stochastic profiling if good single-cell data is available. It should rather be used if single-cells are not measurable and you would like to still like to know something about the underlying heterogeneity that would be lost in bulk measurements. In addition, for very lowly expressed genes more information might be contained in larger pool sizes. This might be seen on data with lower sequencing depth or heterogeneous samples where gene expressions in at least one population are very weak.

7 Summary and Outlook

Biological processes are incredibly complex, so every opportunity for a better understanding of them must be taken. Particularly the evaluation of data generated by biological experiments through mathematical and statistical models contributes to this goal. There, the interaction of disciplines such as biology and mathematics is of utmost importance. Such cooperations (and research in general) work in loops with many iterations: The experimentalist generates data which should be analyzed or modeled through appropriate mathematical methods. Since in general mathematical models are always a simplification of reality, models are only approximations of the ground truth and new questions may arise. In many cases these lead to the need of new experiments and further experimental validation. Through this continuous refinement of knowledge, mathematical modeling contributes to a better understandig of development, interactions and the progress of biological processes (e.g. diseases). In addition, experimental methods are evolving and computing power is increasing over time, which requires a revision and adaption of existing methods.

This thesis summarizes the results of such a process by developing methods, analysing experimental data and readjusting experimental settings: Previously, (Bajikar et al., 2014) have shown that stochastic profiling is suitable to deconvolve measurements of small cell pools to their single-cell profiles. Recent technological achievements now enable us via sequencing to count mRNA numbers directly. In addition, many new experimental protocols for sequencing gene expression of single cells have been established. However, single-cell profiling destroys the natural environment of the cells and affects e. g. cell-to-cell communication which might be preserved when profiling cells together. Results of sequencing experiments contain discrete measurements which in turn should be modeled by discrete distributions rather than continuous distributions. In terms of explainability it is advised to select a distribution that is suitable not only with respect to model estimation but also with respect to interpretability, complexity and biological plausibility of the underlying model assumptions.

We have specifically investigated these developments using the example of AML. In recent years, many new insights on its formation have been gained, but needless to say, the research is not yet complete. Since the AML profiles of different patients differ widely, quantifying the heterogeneity and its evolution over time and thus of the contained subclones is an important research question. Freeze-and-thaw cycles have a huge impact on gene expression profiles which is especially true for AML cells which are known to be very sensitive.

The work this thesis is based on steps further on the way of gaining knowledge. Linking single-cell probability distributions to stochastic processes driving transcription allows us to infer possible corresponding biological transcription models. Although there are several ways to achieve this, we introduce the methodology of Ornstein-Uhlenbeck processes. This general approach of connecting the SDE of the transcription driving process with probability distributions adds a new perspective on transcription models. We support empirically the selection of the negative binomial distribution as a reasonable statistical model for single cell sequencing data. In the future, however, new technologies will certainly emerge that could lead to further conclusions. With the provided toolbox, more distributions and models can be derived. Moreover, stochastic processes will gain additional importance as soon as experimental techniques are developed that generate time-resolved single-cell measurements of mRNA counts. The explanatory value of our models is not affected because noise is not included in the core model. The recent idea of studying gene expression velocity of single-cells (La Manno et al., 2018) is a step in this direction since they present a way to use the gene expression measurements of one time-point and infer its development in the next future by analyzing unspliced mRNA variants which were often ignored during analysis. Therefore a new layer for the splicing of mRNA has to be introduced in the gene expression model. scVelo (Bergen et al., 2020) generalized the method by accounting for stochasticity in gene expression through dynamical modeling.

Further, we determine heterogeneity in pooled cell data by inferring single cell distributions via deconvolution. Integrating the selected NB distribution extends the stochastic profiling algorithm to discrete data. In addition to the revision of the general procedure of deriving single-cell expression profiles from pooling several cells, we are developing further analytical methods. Varying cell numbers in pools and comparing inferred single-cell distributions in different samples is of great use. Often the prediction of population compositions of specific observations is needed to explain specific experimental observations.

Parameter estimation with Bayesian methods allows us to consider not only the measurements as a random variable that follows a distribution, but also the estimated model parameters. Thereby we take into account the uncertainty of the parameter, which is contained in the data and the model. Bayesian computations are becoming more popular as the field of computation evolves and computing power increases even further. Through smart applications of the Bayesian theorem and other approximations computing time and thus costs can be saved.

Discussing noise modeling and heterogeneity detection, it is generally important to define exactly what you are referring to in order not to mix concepts. When using distributions, a homogeneous population shows some variance, which is explained by the univariate distribution. Some (e. g. Brennecke et al., 2015) discover heterogeneity already when there is a large variability, but in our understanding this is not yet heterogeneity but only stochastic variability of gene expression. The same applies to noise. A variable can be regarded as a fixed value, where all deviations can be considered as noise, or as a random variable that follows a distribution and describes the biological variability within a population. Technical and biological variability might not mean anything in terms of real heterogeneity. The analysis of possible noise reduction and heterogeneity detection by cell pooling in both – a presumably homogeneous mESC and an expected heterogeneous AML – datasets of real-world small pool data generated for our purposes gave us valuable insights into the information gain with increasing pool size. However, we could not show that pooling cells substantially reduces noise since the given single-cell measurements already carried less noise than expected. Nevertheless, it must be considered that cell pooling can be advantageous especially if it is not possible to obtain many single cell measurements of high quality. The application of the proposed discrete stochastic profiling algorithm to the further processed datasets showed that heterogeneity can be inferred, but again we could not confirm that larger pool sizes detect more heterogeneity than single-cells. We were able to confirm the presence of heterogeneity in the AML data, but could not derive any further findings.

A possible future application of the stochastic profiling idea lies in the currently growing field of spatial transcriptomics (Asp et al., 2020). There each sequenced spot contains several, but only a small number of cells. These are sequenced together and the location in the tissue is stored. An additional imaging of the slides allows to estimate cell numbers at each spot. Since the entire spot is sequenced together, the gene expression matrix does not contain single-cell measurements, but of small cell pools. Their expression needs to be deconvolved to obtain the contained single-cell expression profiles (Andersson et al., 2020). Note that one of the main assumptions in stochastic profiling is that the cells in one pool are randomly selected. This is not the case in spatial transcriptomics where the cells remain together. This assumption would therefore need to be adjusted.

Abbreviations

ALL	acute lymphoblastic leukemia.
AML	acute myeloid leukemia.
BFMI	estimated Bayesian Fraction of Missing Information.
BIC	Bayesian information criterion.
cDNA	complementary DNA.
CDX	cell line derived xenograft.
CLL	chronic lymphocytic leukemia.
CML	chronic myeloid leukemia.
CRC	colorectal cancer.
ddNTPs	dideoxyribonucleotide triphosphates.
DEL	Delaporte (distribution).
DNA	deoxyribonucleic acid.
dNTPs	deoxyribonucleotide triphosphates.
EXP	exponential (distribution).
FACS	fluorescence-activated cell sorting.
GEMM	genetically engineered mouse model.
Geo	geometric (distribution).
GOF	goodness-of-fit.
HMC	Hamiltonian Monte Carlo.
LN	lognormal (distribution).
MCMC	Markov chain Monte Carlo.
mcSCRB-seq	molecular crowding SCRB-seq.
MDS	myelodysplastic syndrome.
mESC	mouse embryonic stem cells.
ML	maximum likelihood.
MLE	maximum likelihood estimator.
MRD	minimal residual disease.
mRNA	messenger RNA.

NB	negative binomial (distribution).
NGS	next generation sequencing.
NUTS	No-U-Turn sampler.
PB	Poisson-beta (distribution).
PCR	polymerase chain reaction.
PDF	probability density function.
PDX	patient derived xenograft.
PIG	Poisson-inverse Gaussian (distribution).
PMF	probability mass function.
Pois	Poisson (distribution).
pre-mRNA	precursor mRNA.
RNA	ribonucleic acid.
SCRB-seq	single-cell RNA barcoding and sequencing.
scRNA-seq	single-cell RNA sequencing.
snRNA	small nuclear RNA.
STAR	spliced transcripts alignment to a reference.
tRNA	transfer RNA.
UMIs	unique molecular identifiers.

Bibliography

10x Genomics $^{\text{TM}}$ (2018). 10x Genomics Acquires Spatial Transcriptomics.

Abbas, A. R., Wolslegel, K., Seshasayee, D., Modrusan, Z., and Clark, H. F. (2009). Deconvolution of Blood Microarray Data Identifies Cellular Activation Patterns in Systemic Lupus Erythematosus. *PLoS ONE*, 4(7):e6098.

Adan, I. and Resing, J. (2015). *Queueing Systems.* Eindhoven University of Technology Eindhoven.

Aliee, H. and Theis, F. (2020). AutoGeneS: Automatic gene selection using multi-objective optimization for RNA-seq deconvolution. *bioRxiv.*

Amrhein, L. and Fuchs, C. (2020a). Stochastic Profiling of mRNA Counts Using HMC. *Proceedings of the 35th International Workshop on Statistical Modelling (IWSM).*

Amrhein, L. and Fuchs, C. (2020b). stochprofML: Stochastic Profiling Using Maximum Likelihood Estimation in R. *arXiv:2004.08809 [stat.AP].*

Amrhein, L., Harsha, K., and Fuchs, C. (2019). A mechanistic model for the negative binomial distribution of single-cell mRNA counts. *bioRxiv 657619.*

Andersson, A., Bergenstråhle, J., Asp, M., Bergenstråhle, L., Jurek, A., Fernández Navarro, J., and Lundeberg, J. (2020). Single-cell and spatial transcriptomics enables probabilistic inference of cell type topography. *Communications Biology*, 3:565.

Andrews, T. S. and Hemberg, M. (2018). M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics*, 35(16):2865–2867.

Angerer, P., Simon, L., Tritschler, S., Wolf, F. A., Fischer, D., and Theis, F. J. (2017). Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, 4:85 – 91.

Applebaum, D. (2004). *Lévy Processes and Stochastic Calculus.* Cambridge Studies in Advanced Mathematics. Cambridge University Press.

Asp, M., Bergenstråhle, J., and Lundeberg, J. (2020). Spatially Resolved Transcriptomes–Next Generation Tools for Tissue Exploration. *BioEssays*, 42(10):1900221.

Bagnoli, J. W., Ziegenhain, C., Janjic, A., Wange, L. E., Vieth, B., Parekh, S., Geuder, J., Hellmann, I., and Enard, W. (2018). Sensitive and powerful single-cell RNA sequencing using mcSCRB-seq. *Nature Communications*, 9(1):2937.

Bajikar, S. S., Fuchs, C., Roller, A., Theis, F. J., and Janes, K. A. (2014). Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles. *Proceedings of the National Academy of Sciences*, 111(5):E626–E635.

Barndorff-Nielsen, O. E., Jensen, J. L., and Sørensen, M. (1998). Some Stationary Processes in Discrete and Continuous Time. *Advances in Applied Probability*, 30(4):989–1007.

Barndorff-Nielsen, O. E. and Shephard, N. (2001a). Modelling by Lévy Processess for Financial Econometrics. In Barndorff-Nielsen, O. E., Resnick, S. I., and Mikosch, T., editors, *Lévy Processes: Theory and Applications*. Birkhäuser Boston, MA.

Barndorff-Nielsen, O. E. and Shephard, N. (2001b). Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):167–241.

Bengtsson, M. (2005). Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome Research*, 15:1388–1392.

Bennett, J. H. (1845). Case of hypertrophy of the spleen and liver, which death took place from suppuration of the blood. *Edinburgh Med Sug J*, 64:413–423.

Berg, O. G. (1978). A Model for the Statistical Fluctuations of Protein Numbers in a Microbial Population. *Journal of Theoretical Biology*, 71(4):587–603.

Bergen, V., Lange, M., Peidli, S., Wolf, F. A., and Theis, F. J. (2020). Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Boyd, A. L., Aslostovar, L., Reid, J., Ye, W., Tanasijevic, B., Porras, D. P., Shapovalova, Z., Almakadi, M., Foley, R., Leber, B., Xenocostas, A., and Bhatia, M. (2018). Identification of Chemotherapy-Induced Leukemic-Regenerating Cells Reveals a Transient Vulnerability of Human AML Recurrence. *Cancer Cell*, 34(3):483–498.e5.

Brennecke, P., Reyes, A., Pinto, S., Rattay, K., Nguyen, M., Küchler, R., Huber, W., Kyewski, B., and Steinmetz, L. M. (2015). Single-cell transcriptome analysis reveals coordinated ectopic gene-expression patterns in medullary thymic epithelial cells. *Nature Immunology*, 16:933–941.

Brent, R. P. (2010). Unrestricted algorithms for elementary and special functions. *arXiv:1004.3621 [math.NA].*

Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90.

Buettner, F., Natarajan, K. N., Casale, F. P., Proserpio, V., Scialdone, A., Theis, F. J., Teichmann, S. A., Marioni, J. C., and Stegle, O. (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33:155–160.

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36:411–420.

Chen, W., Li, Y., Easton, J., Finkelstein, D., Wu, G., and Chen, X. (2018). UMI-count modeling and differential expression analysis for single-cell RNA sequencing. *Genome Biology*, 19(70).

Colomé-Tatché, M. and Theis, F. (2018). Statistical single cell multi-omics integration. *Current Opinion in Systems Biology*, 7:54–59.

Crick, F. H. C. (1958). On protein synthesis. *Symp. Soc. Exp. Biol., The Biological Replication of Macromolecules*, 12:138–163.

Crick, F. H. C. (1970). Central Dogma of Molecular Biology. *Nature*, 227:561–563.

Darzacq, X., Shav-Tal, Y., de Turris, V., Brody, Y., Shenoy, S. M., Phair, R. D., and Singer, R. H. (2007). In vivo dynamics of RNA polymerase II transcription. *Nature Structural & Molecular Biology*, 14(9):796–806.

Dattani, J. and Barahona, M. (2017). Stochastic models of gene transcription with upstream drives: exact solution and sample path characterization. *Journal of The Royal Society Interface*, 14(126):20160833.

Delmans, M. and Hemberg, M. (2016). Discrete distributional differential expression (D^3E) - a tool for gene expression analysis of single-cell RNA-seq data. *BMC Bioinformatics*, 17(110).

Ding, L., Ley, T. J., Larson, D. E., Miller, C. A., Koboldt, D. C., Welch, J. S., Ritchey, J. K., Young, M. A., Lamprecht, T., McLellan, M. D., McMichael, J. F., Wallis, J. W., Lu, C., Shen, D., Harris, C. C., Dooling, D. J., Fulton, R. S., Fulton, L. L., Chen, K., Schmidt, H., Kalicki-Veizer, J., Magrini, V. J., Cook, L., McGrath, S. D., Vickery, T. L., Wendl, M. C., Heath, S., Watson, M. A., Link, D. C., Tomasson, M. H., Shannon, W. D., Payton, J. E., Kulkarni, S., Westervelt, P., Walter, M. J., Graubert, T. A., Mardis, E. R., Wilson, R. K., and DiPersio, J. F. (2012). Clonal evolution in relapsed acute myeloid leukaemia revealed by whole-genome sequencing. *Nature*, 481:506–510.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21.

Dormann, C. F. (2013). *Parametrische Statistik*. Springer Berlin Heidelberg.

Drăghici, S. (2012). *Statistics and Data Analysis for Microarrays Using R and Bioconductor*. Chapman & Hall/CRC: Mathematical and Computational Biology. Taylor & Francis, 2nd edition.

Ebinger, S., Özdemir, E. Z., Ziegenhain, C., Tiedt, S., Castro Alves, C., Grunert, M., Dworzak, M., Lutz, C., Turati, V. A., Enver, T., Horny, H.-P., Sotlar, K., Parekh, S., Spiekermann, K., Hiddemann, W., Schepers, A., Polzer, B., Kirsch, S., Hoffmann, M., Knapp, B., Hasenauer, J., Pfeifer, H., Panzer-Grümayer, R., Enard, W., Gires, O., and Jeremias, I. (2016). Characterization of Rare, Dormant, and Therapy-Resistant Cells in Acute Lymphoblastic Leukemia. *Cancer Cell*, 30(6):849–862.

Ebinger, S., Zeller, C., Carlet, M., Senft, D., Bagnoli, J. W., Liu, W.-H., Rothenberg-Thurley, M., Enard, W., Metzeler, K. H., Herold, T., Spiekermann, K., Vick, B., and Jeremias, I. (2020). Plasticity in growth behavior of patients' acute myeloid leukemia stem cells growing in mice. *Haematologica*.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10(390).

Erkkilä, T., Lehmusvaara, S., Ruusuvuori, P., Visakorpi, T., Shmulevich, I., and Lähdesmäki, H. (2010). Probabilistic analysis of gene expression measurements from heterogeneous tissues. *Bioinformatics*, 26(20):2571–2577.

Feldman, R. M. and Valdez-Flores, C. (2010). *Applied Probability and Stochastic Processes*. Springer Berlin Heidelberg.

Fenton, L. (1960). The Sum of Log-Normal Probability Distributions in Scatter Transmission Systems. *IEEE Transactions on Communications*, 8(1):57–67.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., Linsley, P. S., and Gottardo, R. (2015). MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(278).

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322.

Frishberg, A., Peshes-Yaloz, N., Cohn, O., Rosentul, D., Steuerman, Y., Valadarsky, L., Yankovitz, G., Mandelboim, M., Iraqi, F. A., Amit, I., Mayo, L., Bacharach, E., and Gat-Viks, I. (2019). Cell composition analysis of bulk genomics using single-cell data. *Nature Methods*, 16:327–332.

Furman, E. (2007). On the convolution of the negative binomial random variables. *Statistics & Probability Letters*, 77(2):169–172.

Gaujoux, R. and Seoighe, C. (2013). CellMix: a comprehensive toolbox for gene expression deconvolution. *Bioinformatics*, 29(17):2211–2212.

Geary, C. G. (2000). The story of chronic myeloid leukaemia. *British Journal of Haematology*, 110:2–11.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis*. Chapman & Hall/CRC: Texts in Statistical Science. Taylor & Francis, 3rd edition.

Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., Emanuelsson, O., Zhang, Z. D., Weissman, S., and Snyder, M. (2007). What is a gene, post-ENCODE? History and updated definition. *Genome Research*, 17(6):669–681.

Gillespie, D. T. (1976). A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *Journal of Computational Physics*, 22(4):403–434.

Goldfarb, D. (1970). A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation*, 24:23–26.

Golding, I., Paulsson, J., Zawilski, S. M., and Cox, E. C. (2005). Real-Time Kinetics of Gene Activity in Individual Bacteria. *Cell*, 123(6):1025–1036.

Gong, T., Hartmann, N., Kohane, I. S., Brinkmann, V., Staedtler, F., Letzkus, M., Bongiovanni, S., and Szustakowski, J. D. (2011). Optimal Deconvolution of Transcriptional Profiling Data Using Quadratic Programming with Application to Complex Clinical Blood Samples. *PLoS ONE*, 6(11):e27156.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351.

Graham, R. L., Knuth, D. E., and Patashnik, O. (2017). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 2nd edition, 31st print edition.

Grün, D., Kester, L., and van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. *Nature Methods*, 11(6):637–640.

Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):296.

Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F., and Theis, F. J. (2016). Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848.

Harries, L. W. (2019). RNA Biology Provides New Therapeutic Targets for Human Disease. *Frontiers in Genetics*, 10:205.

Herold, T., Jurinovic, V., Batcha, A. M. N., Bamopoulos, S. A., Rothenberg-Thurley, M., Ksienzyk, B., Hartmann, L., Greif, P. A., Phillippou-Massier, J., Krebs, S., Blum, H., Amler, S., Schneider, S., Konstandin, N., Sauerland, M. C., Görlich, D., Berdel, W. E., Wörmann, B. J., Tischer, J., Subklewe, M., Bohlander, S. K., Braess, J., Hiddemann, W., Metzeler, K. H., Mansmann, U., and Spiekermann, K. (2018). A 29-gene and cytogenetic score for the prediction of resistance to induction treatment in acute myeloid leukemia. *Haematologica*, 103(3):456–465.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1351–1381.

Holla, M. S. (1967). On a Poisson-Inverse Gaussian Distribution. *Metrika*, 11:115–121.

Howlader, N., Noone, A., Krapcho, M., Miller, D., Brest, A., Yu, M., Ruhl, J., Tatalovich, Z., Mariotto, A., Lewis, D., Chen, H., Feuer, E., and Cronin, K. (1975-2016). *SEER Cancer Statistics Review*. National Cancer Institute. Bethesda, MD. https://seer.cancer.gov/csr/1975_2016/, based on November 2018 SEER data submission, posted to the SEER web site, April 2019.

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J. I., Raj, A., Li, M., and Zhang, N. R. (2018). SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15(7):539–542.

HuBMAP Consortium., Writing Group ., Snyder, M. P. et al. (2019). The human body at cellular resolution: the NIH Human Biomolecular Atlas Program. *Nature*, 574:187–192.

Hunt, G. J., Freytag, S., Bahlo, M., and Gagnon-Bartsch, J. A. (2018). dtangle: accurate and robust cell type deconvolution. *Bioinformatics*, 35(12):2093–2099.

Inman, H. F. and Bradley, E. L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics - Theory and Methods*, 18(10):3851–3874.

Intosalmi, J., Mannerström, H., Hiltunen, S., and Lähdesmäki, H. (2018). SCHiRM: Single Cell Hierarchical Regression Model to detect dependencies in read count data. *bioRxiv 335695*.

Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., and Linnarsson, S. (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166.

Janes, K. A., Wang, C.-C., Holmberg, K. J., Cabral, K., and Brugge, J. S. (2010). Identifying single-cell molecular programs by stochastic profiling. *Nature Methods*, 7(4):311–317.

Jansen, M. and Pfaffelhuber, P. (2015). Stochastic gene expression with delay. *Journal of Theoretical Biology*, 364:355 – 363.

Johnson, W. E., Li, C., and Rabinovic, A. (2006). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127.

Junker, J., Noël, E., Guryev, V., Peterson, K., Shah, G., Huisken, J., McMahon, A., Berezikov, E., Bakkers, J., and van Oudenaarden, A. (2014). Genome-wide RNA Tomography in the Zebrafish Embryo. *Cell*, 159(3):662–675.

Karlis, D. and Xekalaki, E. (2005). Mixed Poisson Distributions. *International Statistical Review*, 73:35–58.

Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742.

Kijima, M. (1997). *Markov Processes for Stochastic Modeling*. Springer, Boston, MA.

Kim, J. K. and Marioni, J. C. (2013). Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome biology*, 14(R7).

Kolodziejczyk, A., Kim, J. K., Svensson, V., Marioni, J., and Teichmann, S. (2015). The Technology and Biology of Single-Cell RNA Sequencing. *Molecular Cell*, 58(4):610–620.

Kurimoto, K. (2006). An improved single-cell cDNA amplification method for efficient high-density oligonucleotide microarray analysis. *Nucleic Acids Research*, 34(5):e42.

Kurz, C. F. (2015). Stochastic profiling of single-cell heterogeneities. Master's thesis, University of Applied Sciences Munich.

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

Lee, P. M. (2012). *Bayesian Statistics: An Introduction.* Wiley Publishing, 4th edition.

Li, W. V. and Li, J. J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature Communications*, 9(997).

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15:1053–1058.

Luecken, M. D. and Theis, F. J. (2019). Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746.

Malone, J. H. and Oliver, B. (2011). Microarrays, deep sequencing and the true measure of the transcriptome. *BMC Biology*, 9(34).

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [stat.ML].*

Muller, K. E. (2001). Computing the confluent hypergeometric function, M(a,b,x). *Numerische Mathematik*, 90:179–196.

Muzzey, D., Evans, E. A., and Lieber, C. (2015). Understanding the Basics of NGS: From Mechanism to Variant Calling. *Current Genetic Medicine Reports*, 3(4):158–165.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Nestorowa, S., Hamey, F. K., Pijuan Sala, B., Diamanti, E., Shepherd, M., Laurenti, E., Wilson, N. K., Kent, D. G., and Gottgens, B. (2016). A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*, 128(8):e20–e31.

Neugebauer, W., Leinberger, D. M., Petersen, K., Schumacher, U., Bachmann, T. T., and Krekel, C. (2010). The Development of a DNA Microarray for the Rapid Identification of Moulds on Works of Art. *Studies in Conservation*, 55(4):258–273.

Newman, A. M., Liu, C. L., Green, M. R., Gentles, A. J., Feng, W., Xu, Y., Hoang, C. D., Diehn, M., and Alizadeh, A. A. (2015). Robust enumeration of cell subsets from tissue expression profiles. *Nature Methods*, 12(5):453–457.

Ng, S. W. K., Mitchell, A., Kennedy, J. A., Chen, W. C., McLeod, J., Ibrahimova, N., Arruda, A., Popescu, A., Gupta, V., Schimmer, A. D., Schuh, A. C., Yee, K. W., Bullinger, L., Herold, T., Görlich, D., Büchner, T., Hiddemann, W., Berdel, W. E., Wörmann, B., Cheok, M., Preudhomme, C., Dombret, H., Metzeler, K., Buske, C., Löwenberg, B., Valk, P. J. M., Zandstra, P. W., Minden, M. D., Dick, J. E., and Wang, J. C. Y. (2016). A 17-gene stemness score for rapid determination of risk in acute leukaemia. *Nature*, 540:433–437.

Olver, F. W. J., Daalhuis, A. B. O., Lozier, D. W., Schneider, B. I., Boisvert, F., Clark, C. W., Miller, B. R., and Saunders, B. V. (2019). *NIST Digital Library of Mathematical Functions.* Release 1.0.22 of 2019-03-15.

Ord, J. K. and Whitmore, G. A. (1986). The poisson-inverse gaussian distribution as a model for species abundance. *Communications in Statistics - Theory and Methods*, 15(3):853–871.

Parekh, S., Ziegenhain, C., Vieth, B., Enard, W., and Hellmann, I. (2018). zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs. *GigaScience*, 7(6).

Pastore, M. and Calcagnì, A. (2019). Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index. *Frontiers in Psychology*, 10:1089.

Paul, F., Arkin, Y., Giladi, A., Jaitin, D., Kenigsberg, E., Keren-Shaul, H., Winter, D., Lara-Astiaso, D., Gury, M., Weiner, A., David, E., Cohen, N., Lauridsen, F., Haas, S., Schlitzer, A., Mildner, A., Ginhoux, F., Jung, S., Trumpp, A., Porse, B., Tanay, A., and Amit, I. (2015). Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell*, 163(7):1663–1677.

Paulsson, J., Berg, O. G., and Ehrenberg, M. (2000). Stochastic focusing: Fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences*, 97(13):7148–7153.

PDQ ® Adult Treatment Editorial Board (2020). PDQ Adult Acute Myeloid Leukemia Treatment, Bethesda, MD: National Cancer Institute. Updated 08/11/2020.

PDQ ® Pediatric Treatment Editorial Board (2020). PDQ Childhood Acute Lymphoblastic Leukemia Treatment, Bethesda, MD: National Cancer Institute. Updated 08/13/2020 .

Pearson, K. (1900). On the Criterion that a given System of Deviations from the Probable in the Case of a Correlated System of Variables is such that it can be reasonably supposed to have arisen from Random Sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Peccoud, J. and Ycart, B. (1995). Markovian Modeling of Gene-Product Synthesis. *Theoretical Population Biology*, 48(2):222–234.

Picelli, S., Faridani, O. R., Björklund, Å. K., Winberg, G., Sagasser, S., and Sandberg, R. (2014). Full-length RNA-seq from single cells using Smart-seq2. *Nature Protocols*, 9:171–181.

Pierson, E. and Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(241).

Qiu, X., Hill, A., Packer, J., Lin, D., Ma, Y.-A., and Trapnell, C. (2017). Single-cell mRNA quantification and differential analysis with Census. *Nature Methods*, 14:309–315.

Raj, A., Peskin, C. S., Tranchina, D., Vargas, D. Y., and Tyagi, S. (2006). Stochastic mRNA Synthesis in Mammalian Cells. *PLoS Biology*, 4(10):e309.

Rajewsky, N., Almouzni, G., Gorski, S. A., et al. (2020). LifeTime and improving European healthcare through cell-based interceptive medicine. *Nature*, 587:377–386.

Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., Campbell, P., Carninci, P., Clatworthy, M., Clevers, H., Deplancke, B., Dunham, I., Eberwine, J., Eils, R., Enard, W., Farmer, A., Fugger, L., Göttgens, B., Hacohen, N., Haniffa, M., Hemberg, M., Kim, S., Klenerman, P., Kriegstein, A., Lein, E., Linnarsson, S., Lundberg, E., Lundeberg, J., Majumder, P., Marioni, J. C., Merad, M., Mhlanga, M., Nawijn, M., Netea, M., Nolan, G., Pe'er, D., Phillipakis, A., Ponting, C. P., Quake, S., Reik, W., Rozenblatt-Rosen, O., Sanes, J., Satija, R., Schumacher, T. N., Shalek, A., Shapiro, E., Sharma, P., Shin, J. W., Stegle, O., Stratton, M., Stubbington, M. J. T., Theis, F. J., Uhlen, M., van Oudenaarden, A., Wagner, A., Watt, F., Weissman, J., Wold, B., Xavier, R., Yosef, N., and Human Cell Atlas Meeting Participants (2017). Science Forum: The Human Cell Atlas. *eLife*, 6:e27041.

Richmond, A. and Su, Y. (2008). Mouse xenograft models vs GEM models for human cancer therapeutics. *Disease Models and Mechanisms*, 1:78–82.

Rigby, R., Stasinopoulos, M., Heller, G., and De Bastiani, F. (2019). *Distributions for Modeling Location, Scale and Shape: Using GAMLSS in R.* Chapman & Hall/CRC: the R series. CRC Press LLC.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9(284).

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47.

Robertson, A., Cipolli, W., and Dascălu, M. (2019). On the Distribution of Monochromatic Complete Subgraphs and Arithmetic Progressions. *Experimental Mathematics.*

Rogers, L. C. G. and Williams, D. (2000). *Diffusions, Markov Processes, and Martingales*, volume 1. Cambridge University Press, Cambridge Mathematical Library, 2nd edition.

Rossi, R. J. (2018). *Mathematical Statistics: An Introduction to Likelihood Based Inference.* John Wiley & Sons, 1st edition.

Sandberg, R. (2014). Entering the era of single-cell transcriptomics in biology and medicine. *Nature Methods*, 11:22–24.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467.

Sato, K.-i. (1999). *Lévy Processes and Infinitely Divisible Distributions.* Number 68 in Cambridge Studies in Advanced Mathematics. Cambridge University Press.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, 270(5235):467–470.

Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., and Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–342.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464.

Shahrezaei, V. and Swain, P. S. (2008). Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105(45):17256–17261.

Shanno, D. F. (1970). Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, 24:647–656.

Shen-Orr, S. S., Tibshirani, R., Khatri, P., Bodian, D. L., Staedtler, F., Perry, N. M., Hastie, T., Sarwal, M. M., Davis, M. M., and Butte, A. J. (2010). Cell type–specific gene expression differences in complex tissues. *Nature Methods*, 7:287–289.

Singh, S., Wang, L., Schaff, D. L., Sutcliffe, M. D., Koeppel, A. F., Kim, J., Onengut-Gumuscu, S., Park, K.-S., Zong, H., and Janes, K. A. (2019). In situ 10-cell RNA sequencing in tissue and tumor biopsy samples. *Scientific Reports*, 9(4836).

Smiley, M. W. and Proulx, S. R. (2010). Gene expression dynamics in randomly varying environments. *Journal of Mathematical Biology*, 61:231–251.

Soumillon, M., Cacchiarelli, D., Semrau, S., van Oudenaarden, A., and Mikkelsen, T. S. (2014). Characterization of directed differentiation by high-throughput single-cell RNA-Seq. *bioRxiv 003236.*

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J. O., Huss, M., Mollbrink, A., Linnarsson, S., Codeluppi, S., Borg, Å., Pontén, F., Costea, P. I., Sahlén, P., Mulder, J., Bergmann, O., Lundeberg, J., and Frisén, J. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82.

Stasinopoulos, M. D., Rigby, R. A., Heller, G. Z., Voudouris, V., and De Bastiani, F. (2017). *Flexible Regression and Smoothing : Using GAMLSS in R*. Chapman & Hall/CRC: the R series. CRC Press LLC.

Stein, C. K., Qu, P., Epstein, J., Buros, A., Rosenthal, A., Crowley, J., Morgan, G., and Barlogie, B. (2015). Removing batch effects from purified plasma cell gene expression microarrays with modified ComBat. *BMC Bioinformatics*, 16(63).

Suter, D. M., Molina, N., Gatfield, D., Schneider, K., Schibler, U., and Naef, F. (2011). Mammalian Genes Are Transcribed with Widely Different Bursting Kinetics. *Science*, 332(6028):472–474.

Tang, W., Bertaux, F., Thomas, P., Stefanelli, C., Saint, M., Marguerat, S., and Shahrezaei, V. (2020). bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics*, 36(4):1174–1181.

Teugels, J. and Sundt, B. (2004). *Encyclopedia of Actuarial Science*, volume 1. Wiley.

Official 10x Genomics Support (2017). https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/hgmm_6k.

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2.

Tietjen, I., Rihel, J. M., Cao, Y., Koentges, G., Zakhary, L., and Dulac, C. (2003). Single-Cell Transcriptional Analysis of Neuronal Progenitors. *Neuron*, 38(2):161–175.

Tirier, S. M., Park, J., Preußer, F., Amrhein, L., Gu, Z., Steiger, S., Mallm, J.-P., Krieger, T., Waschow, M., Eismann, B., Gut, M., Gut, I. G., Rippe, K., Schlesner, M., Theis, F., Fuchs, C., Ball, C. R., Glimm, H., Eils, R., and Conrad, C. (2019). Pheno-seq – linking visual features and gene expression in 3D cell culture systems. *Scientific Reports*, 9(12367).

Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biology*, 20(295).

Valdivieso, L., Schoutens, W., and Tuerlinckx, F. (2009). Maximum likelihood estimation in processes of Ornstein-Uhlenbeck type. *Statistical Inference for Stochastic Processes*, 12:1–19.

Vallejos, C. A., Marioni, J. C., and Richardson, S. (2015). BASiCS: Bayesian Analysis of Single-Cell Sequencing Data. *PLOS Computational Biology*, 11(6):e1004333.

Vick, B., Rothenberg, M., Sandhöfer, N., Carlet, M., Finkenzeller, C., Krupka, C., Grunert, M., Trumpp, A., Corbacioglu, S., Ebinger, M., André, M. C., Hiddemann, W., Schneider, S., Subklewe, M., Metzeler, K. H., Spiekermann, K., and Jeremias,

I. (2015). An Advanced Preclinical Mouse Model for Acute Myeloid Leukemia Using Patients' Cells of Various Genetic Subgroups and In Vivo Bioluminescence Imaging. *PLOS ONE*, 10(3):e0120925.

Vieth, B., Ziegenhain, C., Parekh, S., Enard, W., and Hellmann, I. (2017). powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics*, 33(21):3486–3488.

Vu, T. N., Wills, Q. F., Kalari, K. R., Niu, N., Wang, L., Rantalainen, M., and Pawitan, Y. (2016). Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics*, 32(14):2128–2135.

Wang, N., Hoffman, E. P., Chen, L., Chen, L., Zhang, Z., Liu, C., Yu, G., Herrington, D. M., Clarke, R., and Wang, Y. (2016). Mathematical modelling of transcriptional heterogeneity identifies novel markers and subpopulations in complex tissues. *Scientific Reports*, 6:18909.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10:57–63.

Watson, J. D. and Crick, F. H. C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738.

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15.

Workman, R. E., Tang, A. D., Tang, P. S., Jain, M., Tyson, J. R., Razaghi, R., Zuzarte, P. C., Gilpatrick, T., Payne, A., Quick, J., Sadowski, N., Holmes, N., de Jesus, J. G., Jones, K. L., Soulette, C. M., Snutch, T. P., Loman, N., Paten, B., Loose, M., Simpson, J. T., Olsen, H. E., Brooks, A. N., Akeson, M., and Timp, W. (2019). Nanopore native RNA sequencing of a human poly(A) transcriptome. *Nature Methods*, 16:1297–1305.

Ye, Z.-S. and Chen, N. (2014). The Inverse Gaussian Process as a Degradation Model. *Technometrics*, 56(3):302–311.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(174).

Zha, L., Lord, D., and Zou, Y. (2016). The Poisson Inverse Gaussian (PIG) Generalized Linear Regression Model for Analyzing Motor Vehicle Crash Data. *Journal of Transportation Safety & Security*, 8(1):18–35.

Zhang, Y., Parmigiani, G., and Johnson, W. E. (2020). ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genomics and Bioinformatics*, 2(3):lqaa078.

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(14049).

Ziegenhain, C., Vieth, B., Parekh, S., Hellmann, I., and Enard, W. (2018). Quantitative single-cell transcriptomics. *Briefings in Functional Genomics*, 17(4):220–232.

Ziegenhain, C., Vieth, B., Parekh, S., Reinius, B., Guillaumet-Adkins, A., Smets, M., Leonhardt, H., Heyn, H., Hellmann, I., and Enard, W. (2017). Comparative Analysis of Single-Cell RNA Sequencing Methods. *Molecular Cell*, 65(4):631–643.e4.

A Probability Distributions

A big part of my thesis concentrates on appropriate statistical modeling of data. In practice some measured data needs to be analyzed in order to draw some conclusions. Selecting the right software to analyze given data requires some basic knowledge of the underlying assumptions. For example if the given data is of continuous form only software with underlying models that are applicable to continuous data should be selected. The other way around, if somebody wants to construct some algorithm to analyze some given data, one has to take into account the nature of the data. All models that we use in this thesis are so called parametric models which means they are based on parametric probability distributions. Using parametric models give more power to the result of the analysis. In this part of the Appendix we will introduce all different parametric distributions that will be used in course of the thesis.
Probability distributions and other mathematical terms are often not uniformly defined in literature. In this section, we explain the terminology used in the present work. References include Dormann (2013), the NIST library (Olver et al., 2019), Karlis and Xekalaki (2005), Robertson et al. (2019), Teugels and Sundt (2004) and Graham et al. (2017).

A.1 Continuous Probability Distributions

In this section we present the continuous distributions, that will be used in this thesis. Since a random variable that follows these distributions can take any real value X the probability of X taking one specific value is zero. In general, these distributions are applicable to data with real values, but no value should appear more than once in the data.

Definition A.1 (Lognormal Distribution) *The two parameters defining a univariate lognormal distribution* $\mathrm{LN}(\mu, \sigma^2)$ *are called log-mean* $\mu \in \mathbb{R}$ *and log-standard deviation* $\sigma > 0$. *These are the mean and the standard deviation of the normally distributed random variable* $\log(X)$, *the natural logarithm of* X. *The probability density function*

(PDF) of X is given by

$$f_{\mathrm{LN}}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\log x-\mu)^2}{2\sigma^2}\right) \quad \textit{for } x>0.$$

A random variable $X \sim \mathrm{LN}(\mu,\sigma^2)$ has expectation and variance

$$\mathrm{E}_{\mathrm{LN}}(X) = \exp\left(\mu+\frac{\sigma^2}{2}\right) \quad \textit{and} \quad \mathrm{Var}_{\mathrm{LN}}(X) = \exp\left(2\mu+\sigma^2\right)\left(\exp\left(\sigma^2\right)-1\right). \tag{A.1}$$

Definition A.2 (Inverse Gaussian Distribution) *The inverse Gaussian distribution is a continuous distribution on $[0,\infty)$, parameterized through a mean parameter $\mu>0$ and shape parameter $\lambda>0$ The PDF of X reads*

$$f_{\mathrm{IG}}(x|\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right).$$

A random variable $X \sim \mathrm{IG}(\mu,\lambda)$ has expectation and variance

$$\mathrm{E}_{\mathrm{IG}}[X] = \mu \qquad \textit{and} \qquad \mathrm{Var}_{\mathrm{IG}}[X] = \frac{\mu^3}{\lambda}.$$

The characteristic function is given by

$$\hat{\mu}_X(z) = \exp\left(\frac{\lambda}{\mu}\left(1-\sqrt{1-\frac{2\mu^2 iz}{\lambda}}\right)\right).$$

Definition A.3 (Gamma Distribution) *The gamma distribution is a continuous distribution on $[0,\infty)$, parameterized through a shape parameter $\alpha>0$ and rate parameter $\beta>0$ (which is the inverse of the often-used scale parameter). The PDF of X reads*

$$f_\gamma(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x),$$

where $\Gamma(z)=\int_0^\infty t^{z-1}\exp(-t)dt$ for $z>0$ is the gamma function. A random variable $X \sim \mathrm{Gamma}(\alpha,\beta)$ has expectation and variance

$$\mathrm{E}_\gamma[X] = \frac{\alpha}{\beta} \qquad \textit{and} \qquad \mathrm{Var}_\gamma[X] = \frac{\alpha}{\beta^2}.$$

The characteristic function is given by

$$\hat{\mu}_X(z) = \left(1-\frac{iz}{\beta}\right)^{-\alpha}.$$

For $\alpha=1$, one obtains the exponential distribution. For integer valued α the distribution is also known under the name Erlang distribution.

Definition A.4 (Exponential Distribution) *An exponential distribution* $\mathrm{EXP}(\lambda)$ *is defined by the rate parameter* $\lambda > 0$*. The PDF is given by*

$$f_{\mathrm{EXP}}(x|\lambda) = \lambda \exp(-\lambda x) \quad \textit{for } x \geq 0.$$

A random variable $X \sim \mathrm{EXP}(\lambda)$ *has expectation and variance*

$$\mathrm{E}_{\mathrm{EXP}}[X] = \frac{1}{\lambda} \qquad \textit{and} \quad \mathrm{Var}_{\mathrm{EXP}}[X] = \frac{1}{\lambda^2}.$$

The characteristic function is given by

$$\hat{\mu}_X(z) = \frac{\lambda}{\lambda - iz}$$

Definition A.5 (Beta Distribution) *The standard beta distribution is a continuous distribution on* $(0,1)$*, parameterized through a shape parameter* $\alpha > 0$ *and scale parameter* $\beta > 0$*. The state space can be generalized from* $(0,1)$ *to* (a,c) *by introducing the minimum and maximum values* a *and* c *as additional parameters. The probability density function is*

$$f_\beta(x|\alpha,\beta,a,c) = \frac{(x-a)^{\alpha-1}(c-x)^{\beta-1}}{(c-a)^{\alpha+\beta-1}B(\alpha,\beta)},$$

where $B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt = \Gamma(x+y)/(\Gamma(x)\Gamma(y))$ *for* $x, y > 0$ *is the beta function. A random variable* $X \sim \mathrm{Beta}(\alpha,\beta,a,c)$ *has expectation and variance*

$$\mathrm{E}_\beta[X] = \frac{\alpha c + \beta a}{\alpha + \beta} \qquad \textit{and} \qquad \mathrm{Var}_\beta[X] = \frac{\alpha\beta(c-a)^2}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

The characteristic function of the beta distribution is given by

$$\hat{\mu}_X(z) = \frac{1}{c}\,{}_1F_1(\alpha; \alpha+\beta; iz),$$

where ${}_1F_1$ *is the confluent hypergeometric function of the first kind (see Definition A.6).*

Definition A.6 (Confluent Hypergeometric Function of First Order) *Let* $w, z, a, b \in \mathbb{C}$*. Kummer's equation*

$$z\frac{d^2w}{dz^2} + (b-z)\frac{dw}{dz} - aw = 0$$

has a regular singularity at the origin and an irregular singularity at infinity.One standard solution of this differential equation that only exists if b *is not a non-positive integer is given by the Kummer confluent hypergeometric function* $M(a,b,z)$ *with*

$$M(a,b,z) = \sum_{n=0}^{\infty} \frac{a^{(n)}z^n}{b^{(n)}n!} = {}_1F_1(a;b;z),$$

where ${}_1F_1$ is the confluent hypergeometric function of the first kind with the rising factorial defined through

$$a^{(0)} = 1 \quad \textit{and} \quad a^{(n)} = a(a+1)(a+2)\cdots(a+n-1) = \frac{(a+n-1)!}{(a-1)!} = \frac{\Gamma(a+n)}{\Gamma(a)}.$$

The generalized hypergeometric function is given by

$${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; z) = \sum_{n=0}^{\infty} \frac{a_1^{(n)} \dots a_p^{(n)} z^n}{b_1^{(n)} \dots b_q^{(n)} n!}.$$

If $Re(b) > Re(a) > 0$, $M(a, b, z)$ can be represented as an integral

$$M(a, b, z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{zu} u^{a-1} (1-u)^{b-a-1} \, du.$$

A.2 Discrete Probability Distributions

In this section we will present the discrete probability distributions, that will be used in this thesis. Discrete probability distributions model random variables that can only take on a countable number of values. In contrary to continuous distributions, discrete distributions are tailored to model data with repeating values.

Definition A.7 (Poisson Distribution) *The Poisson* (Pois) *distribution is a discrete count distribution with probability measure*

$$f_{\text{Pois}}(x|\lambda) \equiv \mathrm{P}_{Pois(\lambda)}(X = x) = \frac{\lambda^x}{x!} \exp(-\lambda) \qquad \textit{for } x \in \mathbb{N}_0.$$

The probability generating function of X reads

$$G_{\text{Pois}}(z) = \exp(\lambda(z-1)) \qquad \textit{for } |z| \leq 1$$

and the moment generating function of X is given by

$$M_{\text{Pois}}(t) = \exp(\lambda(e^t - 1)) \qquad \textit{for } t \in \mathbb{R}.$$

A random variable $X \sim \text{Pois}(\lambda)$ has expectation and variance

$$\mathrm{E}_{\text{Pois}}[X] = \lambda \qquad \textit{and} \qquad \mathrm{Var}_{\text{Pois}}[X] = \lambda.$$

Definition A.8 (Negative Binomial Distribution) *The negative binomial* (NB) *distribution is a discrete distribution that describes the probability of an observed number of failures*

$$X \sim \text{NB}(r, p)$$

in a sequence of independent Bernoulli trials until a predefined number of successes has occurred. In each trial, the probability of success is denoted by $p \in [0, 1]$, and the *predefined number of successes is $r \in \mathbb{N}_0$, respectively. The probability mass function of X is given by*

$$f_{\text{NB}}(x|r,p) \equiv \mathrm{P}_{\text{NB}(r,p)}(X=x) = \binom{x+r-1}{x} p^r (1-p)^x \qquad \textit{for } x \in \mathbb{N}_0.$$

The probability generating function of X is given by

$$G_{\text{NB}}(z) = \left(\frac{p}{1-z(1-p)}\right)^r \qquad \textit{for } |z| \leq 1.$$

The above definition of the NB distribution can be extended to $r \in \mathbb{R}_+$. All equations remain valid except for the interpretation in terms of Bernoulli trials. This generalization of r is underpinned by the construction of the Poisson-gamma distribution that is derived along Definition 3.3. Expected value and the variance of NB distributed random variables are given by:

$$\mathrm{E}_{\text{NB}}[X] = \frac{r(1-p)}{p} \qquad \textit{and} \qquad \mathrm{Var}_{\text{NB}}[X] = \frac{r(1-p)}{p^2}.$$

Note: Here, we describe X to represent the number of failures. Literature also provides different parameterizations, where X e. g. denotes the total number of trials (including the last success). The notation used here is the one implemented in the R function nbinom (package stats), with r and p being called size and prob. Another commonly specified parameter is the mean mu of X, given by mu = size/prob − size.

Definition A.9 (Geometric Distribution) *The geometric* (Geo) *distribution is a discrete distribution that describes the probability of*

$$X \sim \mathrm{Geo}(p)$$

failures before the first success in independent Bernoulli trials with success probability p each. The probability mass function of X is given by

$$f_{\text{Geo}}(x|p) \equiv \mathrm{P}_{\text{Geo}(p)}(X=x) = p(1-p)^x \qquad \textit{for } x \in \mathbb{N}_0.$$

A random variable $X \sim \mathrm{Geo}(p)$ has expectation an variance

$$\mathrm{E}_{\text{Geo}}[X] = \frac{1-p}{p} \qquad \textit{and} \qquad \mathrm{Var}_{\text{Geo}}[X] = \frac{1-p}{p^2}$$

Note: $f_{\text{NB}}(x|1,p) \equiv f_{\text{Geo}}(x|p)$.

Definition A.10 (Delaporte Distribution) *The Delaporte* (DEL) *distribution is a discrete distribution with probability measure*

$$\begin{aligned} f_{\text{DEL}}(x|\mu,\sigma,\nu) &\equiv \mathrm{P}_{\text{DEL}(\mu,\sigma,\nu)}(X=x) \\ &= \sum_{j=0}^{x} \frac{\Gamma\left(\frac{1}{\sigma+j}\right)}{j!(y-j)!\Gamma(\frac{1}{\sigma})} \frac{e^{-\mu\nu}(\mu\nu)^{y-j}}{} \left[\mu\sigma(1-\nu)+1\right]^{\frac{1}{\sigma}} \qquad \textit{for } x \in \mathbb{N}_0, \end{aligned}$$

where $\mu, \sigma > 0$ *and* $0 < \nu < 1$. *The probability generating function of* X *reads*

$$G_{\mathrm{DEL}}(z) = \exp(\mu\nu(z-1))[1 + \mu\sigma(1-\nu)(1-z)]^{-\frac{1}{\sigma}} \qquad \textit{for } |z| \leq 1.$$

A random variable $X \sim \mathrm{DEL}(\mu, \sigma, \nu)$ *has expectation and variance*

$$\mathrm{E}_{\mathrm{DEL}}[X] = \mu \qquad \textit{and} \qquad \mathrm{Var}_{\mathrm{DEL}}[X] = \mu + \mu^2\sigma(1-\nu)^2.$$

B Mathematical Identities

This thesis contains many calculations. Some of these derivations need specific mathematical identities, which are listed in the following.

Identity 1 *For the gamma function Γ, one has*

$$\lim_{n\to\infty} \frac{\Gamma(n+\alpha)}{\Gamma(n)n^{\alpha}} = 1, \qquad \alpha \in \mathbb{R}.$$

Next we present some identities involving the binomial series.

Identity 2 *For $|x| < 1$ and r arbitrary real or complex, it holds*

$$\sum_{k=0}^{\infty} \binom{r}{k} x^k = (1+x)^r.$$

Identity 3 *Binomial coefficients are symmetric*

$$\binom{z}{w} = \binom{z}{z-w},$$

with $z \in \mathbb{R} > w \in \mathbb{R} \geq 0$.

Identity 4 *The upper negation of binomial coefficients is given by*

$$\binom{r}{k} = (-1)^k \binom{k-r-1}{k},$$

where k is an integer.

Combining Identities 2,3 and 4 leads to

$$\sum_{k=0}^{\infty} \binom{r+l-1}{r-1} (-x)^l = \sum_{k=0}^{\infty} (-1)^{-l} \binom{-r}{l} (-x)^l = \sum_{k=0}^{\infty} \binom{-r}{l} x^l = \frac{1}{(1+x)^r}. \quad \text{(B.1)}$$

Here, r can be any arbitrary real or complex number but $|x| < 1$.

Identity 5 *The binomial coefficients form together the multinomial coefficient*

$$\begin{aligned}
&\binom{n}{\ell_1}\binom{n-\ell_1}{\ell_2}\cdots\binom{n-\ell_1-\ldots-\ell_{T-2}}{\ell_{T-1}}\\
&\qquad=\frac{n!(n-\ell_1)!\cdots(n-\ell_1-\ldots-\ell_{T-2})!}{\ell_1!\ell_2!...\ell_{T-1}!(n-\ell_1)!(n-\ell_1-\ell_2)!\cdots(n-\ell_1-\ldots-\ell_{T-1})!}=\frac{n!}{\ell_1!\ell_2!\cdots\ell_T!}\\
&\qquad=\binom{n}{\ell_1,\ldots,\ell_T}.
\end{aligned}$$

C Overview of Single-Cell Analysis Tools

Many tools exist that are frequently used in single-cell analysis. In Table 4.1, we provide an overview of those tools that use an underlying probability distribution to describe the counts of a specific gene's mRNA. Most of the tools can be found at `https://www.scrna-tools.org` and at `https://omictools.com`. In the following, we describe the single categories, taken from `www.scrna-tools.org`. Additionally, we added the category *batch correction*.

- Batch Correction: Dealing with data from different batches
- Clustering: Unsupervised grouping of cells based on expression profiles
- Differential Expression: Testing of differential expression across groups of cells
- Dimensionality Reduction: Projection of cells into a lower-dimensional space
- Expression Patterns: Detection of genes that change over a trajectory
- Gene Networks: Identification of co-regulated gene networks
- Gene Sets: Testing or other uses of annotated gene sets
- Imputation: Estimation of expression where zeros have been observed
- Normalization: Removal of unwanted variation that may affect results
- Ordering: Ordering of cells along a trajectory
- Quality Control: Removal of low-quality cells
- Simulation: Generation of synthetic scRNA-seq datasets
- Variable Genes: Identification or use of highly (or lowly) variable genes
- Visualization: Functions for visualizing some aspect of scRNA-seq data or analysis

D Master Equations

In Chapter 4, we show a way how to derive a underlying transcription model from a selected distribution. In order to get there it is important to understand how this is traditionally done the other way around: Using a transcription model and calculating its steady state distribution. To stay consistent in notation and to be complete we included these derivations based on Dattani and Barahona (2017) and Peccoud and Ycart (1995) here in the Appendix.

D.1 Master Equation of the Basic Model

In this section we show in detail how to derive the steady state distribution of the basic model, given in Section 4.2.1. Even though this model is easy to use and steady state distributions can be easily inferred, we will show how to do so in detail. This is particularly important as this builds the basis for all following sections. First we will show how to set up the master equation and determine the steady state distribution of the mRNA counts.

$\mathcal{P}(,t)$ describes the probability of having n M mRNAs at time t in the system. The (chemical) master equation that we use to build the stochastic model, is easily set up by looking at the different events that can happen in Δt. We assume that at maximum one event can occur during this short time interval and that the reaction probability is the reaction propensity times Δt. The reaction propensity for transcription is fixed and given by r_{tran}. In contrast, the reaction propensity for degradation is dependent on the number of mRNAs currently available. In order to have n mRNAs after a degradation event there must have been $n+1$ mRNAs before and thus the propensity is given by $r_{deg}(n+1)$. Teken together, in the basic model in one time interval either a molecule is generated, degrades or nothing happens.

$$\begin{aligned}
\mathcal{P}(n,t) = &\mathcal{P}(n-1,t-\Delta t)\mathcal{P}(\text{“mRNA is transcribed”})\\
&+\mathcal{P}(n+1,t-\Delta t)\mathcal{P}(\text{“mRNA degrades”})\\
&+\mathcal{P}(n,t-\Delta t)\mathcal{P}(\text{“no event”})
\end{aligned}$$

Filling in the probabilities of the different scenarios leads to

$$\begin{aligned}
=&\mathcal{P}(n-1,t-\Delta t)(r_{tran}\Delta t+o(\Delta t))\\
&+\mathcal{P}(n+1,t-\Delta t)(r_{deg}(n+1)\Delta t+o(\Delta t))\\
&+\mathcal{P}(n,t-\Delta t)(1-r_{tran}\Delta t-r_{deg}\,n\Delta t+o(\Delta t)).
\end{aligned}$$

Subtract $\mathcal{P}\,(n,t-\Delta t)$ on both sides

$$\begin{aligned}
\mathcal{P}(n,t)-\mathcal{P}\,(n,t-\Delta t)=&(r_{tran}\Delta t+o(\Delta t))\mathcal{P}(n-1,t-\Delta t)\\
&+(r_{deg}(n+1)\Delta t+o(\Delta t))\mathcal{P}(n+1,t-\Delta t)\\
&-(r_{tran}\Delta t+r_{deg}\,n\Delta t+o(\Delta t))\mathcal{P}(n,t-\Delta t).
\end{aligned}$$

Divide by Δt

$$\begin{aligned}
\frac{\mathcal{P}(n,t)-\mathcal{P}(n,t-\Delta t)}{\Delta t}=&\frac{(r_{tran}\Delta t+o(\Delta t))}{\Delta t}\mathcal{P}(n-1,t-\Delta t)\\
&+\frac{(r_{deg}(n+1)\Delta t+o(\Delta t))}{\Delta t}\mathcal{P}(n+1,t-\Delta t)\\
&-\frac{(r_{tran}\Delta t+r_{deg}\,n\Delta t+o(\Delta t))}{\Delta t}\mathcal{P}(n,t-\Delta t)\\
=&\left(r_{tran}+\frac{o(\Delta t)}{\Delta t}\right)\mathcal{P}(n-1,t-\Delta t)\\
&+\left(r_{deg}(n+1)+\frac{o(\Delta t)}{\Delta t}\right)\mathcal{P}(n+1,t-\Delta t)\\
&-\left(r_{tran}+r_{deg}\,n+\frac{o(\Delta t)}{\Delta t}\right)\mathcal{P}(n,t-\Delta t).
\end{aligned}$$

Let $\Delta t\to 0$

$$\frac{d\mathcal{P}(n,t)}{dt}=r_{tran}\mathcal{P}(n-1,t)+r_{deg}(n+1)\mathcal{P}(n+1,t)-(r_{tran}+r_{deg}\,n)\mathcal{P}(n,t)$$

With help of the probability generating function

$$G(z,t)=\sum_{n=0}^{\infty}z^n\mathcal{P}(n,t|r_{tran},r_{deg}),\tag{D.1}$$

we get its partial derivatives

$$\begin{aligned}
\frac{\partial G}{\partial z}(z,t)=&\sum_{n=0}^{\infty}n\,z^{(n-1)}\mathcal{P}(n,t|r_{tran},r_{deg})\\
\frac{\partial G}{\partial t}(z,t)=&\sum_{n=0}^{\infty}z^n\frac{d\mathcal{P}(n,t|r_{tran},r_{deg})}{\mathrm{d}t}\\
=&\sum_{n=0}^{\infty}z^n\left(r_{tran}\mathcal{P}(n-1,t)+r_{deg}(n+1)\mathcal{P}(n+1,t)-(r_{tran}+r_{deg}\,n)\mathcal{P}(n,t)\right)
\end{aligned}$$

$$
\begin{aligned}
&= r_{tran}\, z \sum_{n=0}^{\infty} z^{n-1}\mathcal{P}(n-1,t) + r_{deg} \sum_{n=0}^{\infty} z^{n}(n+1)\mathcal{P}(n+1,t) \\
&\quad - r_{tran} \sum_{n=0}^{\infty} z^{n}\mathcal{P}(n,t) - r_{deg}\, z \sum_{n=0}^{\infty} z^{n-1} n\mathcal{P}(n,t) \\
&= r_{tran}\, zG(z,t) + r_{deg} \frac{\partial G}{\partial z}(z,t) - r_{tran} G(z,t) - r_{deg}\, z \frac{\partial G}{\partial z(z,t)}.
\end{aligned}
$$

The resulting partial differential equation (PDE) is given by

$$
\frac{\partial G}{\partial t}(z,t) =(z-1) r_{tran} G(z,t) - (z-1) r_{deg} \frac{\partial G}{\partial z}(z,t). \tag{D.2}
$$

The solution of (D.2) with initial condition of n_0 molecules is calculated by using the methods of characteristics, where u is the characteristic curve and without loss of generality we let $t(0) = 0$ and reparametrize

$$
\frac{dG}{du} =\frac{\partial G}{\partial z}\frac{dz}{du} + \frac{\partial G}{\partial t}\frac{dt}{ds},
$$

so that

$$
\frac{dz}{du} = (z-1) r_{deg} \qquad \frac{dt}{du} = 1 \qquad \text{and} \qquad \frac{dG}{du} = (z-1) r_{tran} G.
$$

Next, we solving these:

$$
\begin{aligned}
\log(z-1) - \log(z_0-1) &= \int_0^s r_{deg} dx \\
(z-1) &= (z_0-1)\exp\left(\int_0^s r_{deg} dx\right)
\end{aligned} \tag{D.3}
$$

and

$$
t = s.
$$

Furthermore

$$
\begin{aligned}
\log(G) - \log(G_0) &= \int_0^s (z-1) r_{tran} d\tau \\
G &= G_0 \exp\left(\int_0^s (z-1) r_{tran} d\tau\right) = G_0 \exp\left(\int_0^s r_{tran}(z_0-1)\exp\left(\int_0^\tau r_{deg} dx\right) d\tau\right). \\
&= G_0 \exp\left((z_0-1)\int_0^s r_{tran}\exp\left(\int_0^\tau r_{deg} dx\right) d\tau\right)
\end{aligned} \tag{D.4}
$$

With $P(n_0,0) = 1$ it follows that $G_0 = G(z_0,0) = (z_0)^{n_0}$ and with (D.3) it follows that $z_0 = (z-1)\exp\left(-\int_0^t r_{deg} dx\right) + 1$. Therfore

$$
G(z,t|n_0) = \left[(z-1)\exp\left(-\int_0^t r_{deg} dx\right) + 1\right]^{n_0}
$$

$$\exp\left((z-1)\exp\left(-\int_0^s r_{deg}dx\right)\int_0^s r_{tran}\exp\left(\int_0^\tau r_{deg}dx\right)d\tau\right)$$
$$=\left[(z-1)e^{-\int_0^t r_{deg}dx}+1\right]^{n_0}e^{(z-1)\int_0^s r_{tran}e^{-\int_\tau^s r_{deg}dx}d\tau}.$$

This can be writen as:

$$G(z,t|n_0)=\left[(z-1)e^{-r_{deg}t}+1\right]^{n_0}e^{I(t)(z-1)},\ \text{with } I(t)=\int_0^t r_{tran}e^{-\int_\tau^t r_{deg}d\tau'}d\tau$$

The first part is the distribution of the starting condition, i. e. the number of molecules at the beginning n_0 and the second part corresponds to the long term distribution of the molecules content. The second part is the time dependent probability generating function of a compound Poisson distribution with time dependent intensity parameter $I(t)$ (see Definition 3.3). Note that here, we have a constant rates r_{tran} and r_{deg} but we treated them as they were not. We have done this so that the calculation can also be reused for all other processes where the rates are not constant.
Therefore in this model $I(t)$ can be further simplified:

$$I(t)=\int_0^t r_{tran}e^{-\int_\tau^t r_{deg}d\tau'}d\tau=\int_0^t r_{tran}e^{-r_{deg}(t-\tau)}d\tau$$
$$=r_{tran}e^{-r_{deg}t}\int_0^t r_{tran}e^{r_{deg}\tau}d\tau=r_{tran}e^{-r_{deg}t}\left(\frac{e^{r_{deg}t}-1}{r_{deg}}\right)=\frac{r_{tran}}{r_{deg}}\left(1-e^{-r_{deg}t}\right)$$

For $t\to\infty$ the steady state distribution of molecules is independent of the starting material of molecules and of time and hence is a Poisson distribution with time-independent, constant intensity parameter $I=\frac{r_{tran}}{r_{deg}}$.
The corresponding calculations can also be performed from the perspective of queueing systems (see Appendix E.1).

D.2 Master Equation of the Generalized Model

Next, we describe the derivation of steady-state distributions for mRNA counts in the generalized model. In the following, $\mathcal{P}(n,t)$ describes the probability of having n mRNA molecules at time t in the system. The derivation is completely similar to the one for the basic model. The master equation is set up by looking at the reactions (at most one) that can happen within an infinitesimally small time interval: Either one mRNA molecule is transcribed, which happens with probability rate R_t, or one mRNA molecule degrades with rate r_{deg}, or nothing happens. In the following, we write $\mathcal{P}(n,t|R_t,r_{deg})=\mathcal{P}(n,t)$ for the sake of simpler notation. The master equation reads

$$\frac{d\mathcal{P}(n,t)}{dt}=R_t\mathcal{P}(n-1,t)+r_{deg}(n+1)\mathcal{P}(n+1,t)-(R_t+r_{deg}\,n)\mathcal{P}(n,t).$$

Again analog to the basic model the probability generating function

$$G(z,t)=\sum_{n=0}^{\infty}z^n\mathcal{P}(n,t)$$

and its partial derivatives are obtained

$$\frac{\partial G}{\partial z}(z,t) = \sum_{n=0}^{\infty} n\, z^{(n-1)} \mathcal{P}(n,t)$$

and

$$\begin{aligned}
\frac{\partial G}{\partial t}(z,t) =& \sum_{n=0}^{\infty} z^n \frac{d\mathcal{P}(n,t)}{\mathrm{d}t} \\
=& \sum_{n=0}^{\infty} z^n \left(R_t \mathcal{P}(n-1,t) + r_{deg}(n+1)\mathcal{P}(n+1,t) - (R_t + r_{deg}\, n)\mathcal{P}(n,t) \right) \\
=& R_t\, z \sum_{n=0}^{\infty} z^{n-1} \mathcal{P}(n-1,t) + r_{deg} \sum_{n=0}^{\infty} z^n (n+1) \mathcal{P}(n+1,t) \\
& \quad - R_t \sum_{n=0}^{\infty} z^n \mathcal{P}(n,t) - r_{deg}\, z \sum_{n=0}^{\infty} z^{n-1} n \mathcal{P}(n,t) \\
=& R_t\, z G(z,t) + r_{deg} \frac{\partial G}{\partial z}(z,t) - R_t G(z,t) - r_{deg}\, z \frac{\partial G}{\partial z}(z,t).
\end{aligned}$$

and results in the PDE

$$\frac{\partial G}{\partial t}(z,t) = (z-1) R_t G(z,t) - (z-1) r_{deg} \frac{\partial G}{\partial z}(z,t).$$

that is solved with initial condition of having n_0 mRNA molecules by using the methods of characteristics:

$$G(z,t|n_0) = \left[(z-1)e^{-r_{deg}t} + 1 \right]^{n_0} e^{I_t(z-1)} \qquad \text{with } I_t = \int_0^t R_\tau e^{-\int_\tau^t r_{deg} d\tau'} d\tau.$$

The first factor of $G(z,t|n_0)$ reflects the dependence of the distribution on the initial value n_0. The second factor $\exp(I_t(z-1))$ corresponds to the long-term behaviour of the mRNA content and equals the time-dependent probability generating function of a Poisson distribution with intensity parameter I_t (see Definition 3.3). One commonly considers the distribution in steady state (if that state exists), meaning $t \to \infty$. In this limit, the first factor vanishes (i. e. becomes one). Thus, the steady-state distribution is independent of the starting condition. The second term remains. Thus, in steady state the mRNA count follows a conditional Poisson distribution with intensity parameter I_t being governed by the transcription and degradation process. From Definition 3.3, one gets

$$\mathcal{P}_{\text{steady state}}(n,t) = \mathcal{P}_{I_t}(n,t) = \int_0^\infty \frac{x^n}{n!} e^{-x} f_{I_t}(x,t) dx$$

D.3 Masterequation of the Switching Model

In this part of the Appendix, we follow the calculations of Dattani and Barahona (2017), Smiley and Proulx (2010) and Raj et al. (2006) who show how to derive the steady state distribution of mRNA content in the switching model (Figure 4.3). As written in the main text, the RDE (4.5) now reads

$$\mathrm{d}I_t = -r_{deg} I_t \mathrm{d}t + r_{switch}(t)\mathrm{d}t.$$

As transcription is governed by a Markov process which is a random process and not deterministic anymore, the probability distribution for the amount of mRNA at time t is a compound Poisson distribution as described by (4.4). Again, in order to determine the steady-state distribution of mRNA counts, the steady-state distribution of I_t in (4.8) needs to be determined. The Markov process $r_{switch}(t)$ can be characterized by its infinitesimal generator

$$Q = \begin{bmatrix} -r_{act} & r_{deact} \\ r_{act} & -r_{deact} \end{bmatrix},$$

where the entries on the anti-diagonal Q_{ij} $(i \neq j)$ are the transition rate constants from state j to i and its reciprocals are the means of the exponential waiting times. States 1 and 2 correspond to the inactive and the active state, respectively. This means r_{act} corresponds to the rate with which a gene is activated (transition from state 1 to 2), and r_{deact} is the deactivation rate, that is the rate of the transition from state 2 to 1. The probability transition matrix $P(t)$ is defined as

$$P(t) = \frac{1}{r_{act} + r_{deact}} \begin{bmatrix} r_{deact} + r_{act} e^{-(r_{deact}+r_{act})t} & r_{deact} - r_{deact} e^{-(r_{deact}+r_{act})t} \\ r_{act} - r_{act} e^{-(r_{deact}+r_{act})t} & r_{act} + r_{deact} e^{-(r_{deact}+r_{act})t} \end{bmatrix}.$$

$P(t)$ satisfies the Kolmogorov differential equation $P'(t) = QP(t)$, and the initial condition is

$$P(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The entry $P_{ij}(t)$ denotes the probability of a transition from state j to i. (Note: Here, Q and $P(t)$ are the transpose of the usual notation as this notation is more convenient in the present stationary analysis.) If the probabilities for $r_{switch}(0)$ being in state 1 or 2 are given by $p(0) = [p_{\mathrm{off}}(0), p_{\mathrm{on}}(0)]^T$, then the distribution of $r_{switch}(t)$ is given by $p(t) = P(t)p(0)$ and it follows that

$$p(t) = \frac{1}{r_{act} + r_{deact}} \begin{bmatrix} r_{deact} + (r_{act} p_{off}(0) - r_{deact} p_{on}(0)) e^{-(r_{deact}+r_{act})t} \\ r_{act} + (r_{deact} p_{on}(0) - r_{act} p_{off}(0)) e^{-(r_{deact}+r_{act})t} \end{bmatrix}. \tag{D.5}$$

The vector $p(t)$ has to fulfill the Kolmogorov differential equation

$$p'(t) = Qp(t) \tag{D.6}$$

as well. Assume $0 \leq r_{off} < r_{on}$, then $I_0 \in [r_{off}/r_{deg}, r_{on}/r_{deg}]$ and, with probability one, one has $I_t \in [r_{off}/r_{deg}, r_{on}/r_{deg}]$ for $t > 0$. One has

$$\mathcal{P}(I_t \in [x, x + \triangle x]) = \mathcal{P}(I_t \in [x, x + \triangle x], r_{switch}(t) = r_{on})$$

$$+\,\mathcal{P}(I_t \in [x, x+\triangle x], r_{switch}(t) = r_{off}).$$

The joint cumulative distribution functions (CDFs) associated with the joint probabilities of I_t being equal to x and $r_{switch}(t)$ being equal to r_i are given by

$$\Psi_i(x,t) = \mathcal{P}(I_t \leq x, r_{switch}(t) = r_i), \qquad \text{for } x \geq 0 \text{ and } i \in \{\text{on}, \text{off}\}.$$

Their derivatives with respect to x given the joint distribution of $I_t = x$ and $r_{switch}(t) = r_i$ is denoted as $\psi_i(x,t)$. The probability density function (PDF) $\psi(x,t)$ associated with I_t can be characterized by a system of two PDEs

$$\psi(x,t) = \psi_{on}(x,t) + \psi_{off}(x,t), \quad x \in \left[\frac{r_{off}}{r_{deg}}, \frac{r_{on}}{r_{deg}}\right].$$

Clearly, with (D.5) one obtains

$$\int_{r_{off}/r_{deg}}^{r_{on}/r_{deg}} \psi_i(x,t)dx = \mathcal{P}\left(I_t \in \left[\frac{r_{off}}{r_{deg}}, \frac{r_{on}}{r_{deg}}\right], r_{switch}(t) = r_i\right) = p_i(t), \qquad i \in \{\text{on}, \text{off}\}. \tag{D.7}$$

We now set

$$q(x,t) = \begin{bmatrix} \psi_{off}(x,t) \\ \psi_{on}(x,t) \end{bmatrix},$$

which is still directly connected with the two-state Markov process $r_{switch}(t)$. Both components of $q(x,t)$ are continuous PDFs, one for each state of $r_{switch}(t)$. This is again a two-state Markov process and adopts the transition rate matrix Q from the process $r_{switch}(t)$. It hence inherits its property (D.6), and thus, $q(x,t)$ fulfills the Kolmogorov differential equation as well, i. e.

$$q'(x,t) = Qq(x,t).$$

All together

$$\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{dt}}\psi_{off}(x,t) \\ \frac{\mathrm{d}}{\mathrm{dt}}\psi_{on}(x,t) \end{bmatrix} = \begin{bmatrix} -r_{act} & r_{deact} \\ r_{act} & -r_{deact} \end{bmatrix} \begin{bmatrix} \psi_{off}(x,t) \\ \psi_{on}(x,t) \end{bmatrix}$$

and thus

$$\begin{bmatrix} \frac{\partial}{\partial t}\psi_{off}(x,t) + \frac{\partial}{\partial x}\psi_{off}(x,t)\frac{\mathrm{dx}}{\mathrm{dt}} \\ \frac{\partial}{\partial t}\psi_{on}(x,t) + \frac{\partial}{\partial x}\psi_{on}(x,t)\frac{\mathrm{dx}}{\mathrm{dt}} \end{bmatrix} = \begin{bmatrix} -r_{act}\psi_{off}(x,t) + r_{deact}\psi_{on}(x,t) \\ r_{act}\psi_{off}(x,t) - r_{deact}\psi_{on}(x,t) \end{bmatrix}.$$

Using (4.8), we get $\frac{\mathrm{dx}}{\mathrm{dt}} = -r_{deg}x + r_{switch}(t)$. Plugging this in, the system of PDEs can be simplified to

$$\frac{\partial}{\partial t}\psi_{off}(x,t) + \frac{\partial}{\partial x}[\psi_{off}(x,t)(r_{off} - r_{deg}x)] = -r_{act}\psi_{off}(x,t) + r_{deact}\psi_{on}(x,t) \tag{D.8}$$

$$\frac{\partial}{\partial t}\psi_{on}(x,t) + \frac{\partial}{\partial x}[\psi_{on}(x,t)(r_{on} - r_{deg}x)] = r_{act}\psi_{\mathit{off}}(x,t) - r_{deact}\psi_{on}(x,t), \qquad \text{(D.9)}$$

which correspond to Equations (6) in Smiley and Proulx (2010). Integrating both sides of (D.8) and (D.9) with respect to x over the range from r_{off}/r_{deg} to r_{on}/r_{deg} leads us to

$$\begin{aligned}&\frac{\partial}{\partial t}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{\mathit{off}}(x,t)\mathrm{d}x + \int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \frac{\partial}{\partial x}[\psi_{\mathit{off}}(x,t)/r_{\mathit{off}} - r_{deg}x)]\mathrm{d}x\\ =& - r_{act}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{\mathit{off}}(x,t)\mathrm{d}x + r_{deact}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{on}(x,t)\mathrm{d}x\end{aligned}$$

and

$$\begin{aligned}&\frac{\partial}{\partial t}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{on}(x,t)\mathrm{d}x + \int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \frac{\partial}{\partial x}[\psi_{on}(x,t)(r_{on} - r_{deg}x)]\mathrm{d}x\\ =& r_{act}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{off}(x,t)\mathrm{d}x - r_{deact}\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} \psi_{on}(x,t)\mathrm{d}x.\end{aligned}$$

With (D.7), it follows that

$$\frac{\partial}{\partial t}p_{\mathit{off}}(t) + [\psi_{\mathit{off}}(x,t)(r_{\mathit{off}} - r_{deg}x)]_{r_{\mathit{off}}/r_{deg}}^{r_{on}/r_{deg}} = -r_{act}p_{\mathit{off}}(t) + r_{deact}p_{on}(t)$$

and

$$\frac{\partial}{\partial t}p_{on}(t) + [\psi_{on}(x,t)(r_{on} - r_{deg}x)]_{r_{\mathit{off}}/r_{deg}}^{r_{on}/r_{deg}} = r_{act}p_{\mathit{off}}(t) - r_{deact}p_{on}(t).$$

Since Equation (D.6) still has to be fulfilled, it follows directly that the redundant terms have to be equal to zero:

$$\psi_{\mathit{off}}\left(\frac{r_{on}}{r_{deg}},t\right)(r_{\mathit{off}} - r_{on}) + \psi_{\mathit{off}}\left(\frac{r_{\mathit{off}}}{r_{deg}},t\right)(r_{\mathit{off}} - r_{\mathit{off}}) \overset{!}{=} 0,$$

which is equivalent to

$$\psi_{\mathit{off}}\left(\frac{r_{on}}{r_{deg}},t\right) = 0 \qquad \text{for } t > 0.$$

Similarly,

$$\psi_{on}\left(\frac{r_{on}}{r_{deg}},t\right)(r_{on} - r_{on}) - \psi_{on}\left(\frac{r_{\mathit{off}}}{r_{deg}},t\right)(r_{on} - r_{\mathit{off}}) \overset{!}{=} 0,$$

which implies

$$\psi_{on}\left(\frac{r_{\mathit{off}}}{r_{deg}},t\right) = 0 \qquad \text{for } t > 0. \qquad \text{(D.10)}$$

Following Smiley and Proulx (2010), the PDF of the stationary distribution of $\psi(x,t)$, denoted by f_{I_t}, which is analogously determined by a pair of functions $f_{I_t,\mathit{off}}$ and $f_{I_t,on}$ is investigated via

$$f_{I_t}(x) = f_{I_t,\mathit{off}}(x) + f_{I_t,on}(x),$$

with $f_{I_t,\mathit{off}}$ and $f_{I_t,on}$ being the time-independent solutions of (D.8) and (D.9). Those can be calculated by solving the time-independent versions of (D.8) and (D.9), given by

$$\frac{\mathrm{d}}{\mathrm{d}x}[f_{I_t,\mathit{off}}(x)(r_{\mathit{off}} - r_{deg}x)] = -r_{act}f_{I_t,\mathit{off}}(x) + r_{deact}f_{I_t,on}(x) \tag{D.11}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[f_{I_t,on}(x)(r_{on} - r_{deg}x)] = r_{act}f_{I_t,\mathit{off}}(x) - r_{deact}f_{I_t,on}(x) \tag{D.12}$$

with integral conditions derived from Equation (D.7) for $t \to \infty$

$$\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} f_{I_t,\mathit{off}}(x)\mathrm{d}x = \frac{r_{deact}}{r_{act} + r_{deact}}, \tag{D.13}$$

$$\int_{\frac{r_{\mathit{off}}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} f_{I_t,on}(x)\mathrm{d}x = \frac{r_{act}}{r_{act} + r_{deact}}. \tag{D.14}$$

Summing up (D.11) and (D.12) results in

$$\frac{\mathrm{d}}{\mathrm{d}x}[f_{I_t,\mathit{off}}(x)(r_{\mathit{off}} - r_{deg}x) + f_{I_t,on}(x)(r_{on} - r_{deg}x)] = 0 \qquad \text{for} \quad \frac{r_{\mathit{off}}}{r_{deg}} < x < \frac{r_{on}}{r_{deg}}.$$

For any solution of (D.11) and (D.12) and for any constant K it follows that

$$f_{I_t,\mathit{off}}(x)(r_{\mathit{off}} - r_{deg}x) + f_{I_t,on}(x)(r_{on} - r_{deg}x) = K \qquad \text{for} \quad \frac{r_{\mathit{off}}}{r_{deg}} < x < \frac{r_{on}}{r_{deg}},$$

thus

$$f_{I_t,on}(x) = \frac{(r_{deg}x - r_{\mathit{off}})f_{I_t,\mathit{off}}(x) + K}{r_{on} - r_{deg}x}. \tag{D.15}$$

Plugging in (D.15) into (D.11) and setting $K = 0$ (as all steady-state solutions have to satisfy the condition given in (D.10)), one gets

$$f'_{I_t,\mathit{off}}(x) = \left(-\frac{r_{act}}{r_{\mathit{off}} - r_{deg}x} - \frac{r_{deact}}{r_{on} - r_{deg}x} + \frac{r_{deg}}{r_{\mathit{off}} - r_{deg}x}\right) f_{I_t,\mathit{off}}(x),$$

which can be solved up to a normalizing factor C:

$$\begin{bmatrix} f_{I_t,\mathit{off}}(x) \\ f_{I_t,on}(x) \end{bmatrix} = C \begin{bmatrix} (r_{deg}x - r_{\mathit{off}})^{\frac{r_{act}}{r_{deg}}-1}(r_{on} - r_{deg}x)^{\frac{r_{deact}}{r_{deg}}} \\ (r_{deg}x - r_{\mathit{off}})^{\frac{r_{act}}{r_{deg}}}(r_{on} - r_{deg}x)^{\frac{r_{deact}}{r_{deg}}-1} \end{bmatrix}.$$

Equations (D.13) and (D.14) are used to determine C:

$$\begin{aligned}\int_{\frac{r_{off}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} & \big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}-1}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}}\,\mathrm{d}x \\ &= \frac{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}}}{r_{deg}} B\left(\frac{r_{act}}{r_{deg}}, 1 + \frac{r_{deact}}{r_{deg}}\right) \\ &= \frac{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}}}{r_{deg}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right) \frac{r_{deact}}{r_{act} + r_{deact}}\end{aligned}$$

and

$$\begin{aligned}\int_{\frac{r_{off}}{r_{deg}}}^{\frac{r_{on}}{r_{deg}}} & \big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}-1}\mathrm{d}x \\ &= \frac{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}}}{r_{deg}} B\left(1 + \frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right) \\ &= \frac{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}}}{r_{deg}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right) \frac{r_{act}}{r_{act} + r_{deact}}.\end{aligned}$$

Here, B denotes the beta function as introduced in Definition A.5. Both of the above integrals have to be normalized by

$$\frac{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}}}{r_{deg}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)$$

in order to result in $r_{deact}/(r_{act} + r_{deact})$ as given by (D.13) and $r_{act}/(r_{act} + r_{deact})$ as given by (D.14), respectively. All together, this results into

$$\begin{aligned}f_{I_t,off}(x) &= \frac{r_{deg}\big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}-1}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}}}{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)} \\ f_{I_t,on}(x) &= \frac{r_{deg}\big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}-1}}{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)}.\end{aligned}$$

Adding these up will provide the final solution

$$\begin{aligned}f_{I_t}(x) &= f_{I_t,on}(x) + f_{I_t,off}(x) \\ &= \frac{r_{deg}\big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}-1}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}-1}\big[\big(r_{on} - r_{deg}x\big) + \big(r_{deg}x - r_{off}\big)\big]}{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)} \\ &= \frac{r_{deg}\big(r_{deg}x - r_{off}\big)^{\frac{r_{act}}{r_{deg}}-1}\big(r_{on} - r_{deg}x\big)^{\frac{r_{deact}}{r_{deg}}-1}}{\big(r_{on} - r_{off}\big)^{\frac{r_{act}+r_{deact}}{r_{deg}}-1} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)}\end{aligned}$$

$$
\begin{aligned}
&= \frac{r_{deg}^{1+\frac{r_{act}}{r_{deg}}-1} \left(x - \frac{r_{\mathit{off}}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}-1} r_{deg}^{\frac{r_{deact}}{r_{deg}}-1} \left(\frac{r_{on}}{r_{deg}} - x\right)^{\frac{r_{deact}}{r_{deg}}-1}}{\left(r_{on} - r_{\mathit{off}}\right)^{\frac{r_{act}+r_{deact}}{r_{deg}}-1} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)} \\
&= \frac{\left(x - \frac{r_{\mathit{off}}}{r_{deg}}\right)^{\frac{r_{act}}{r_{deg}}-1} \left(\frac{r_{on}}{r_{deg}} - x\right)^{\frac{r_{deact}}{r_{deg}}-1}}{\left(\frac{r_{on}}{r_{deg}} - \frac{r_{\mathit{off}}}{r_{deg}}\right)^{\frac{r_{act}+r_{deact}}{r_{deg}}-1} B\left(\frac{r_{act}}{r_{deg}}, \frac{r_{deact}}{r_{deg}}\right)}. \qquad \text{(D.16)}
\end{aligned}
$$

This is the density of the stationary distribution of I_t from Equation (4.8), and it is the density function of a four-parametric beta distribution (see Definition A.5) with parameters $a = r_{\mathit{off}}/r_{deg}$, $c = r_{on}/r_{deg}$, $\alpha = r_{act}/r_{deg}$ and $\beta = r_{deact}/r_{deg}$.

E Queueing Systems

All birth-death processes can be linked to queueing systems, where the process is understood as a system where customers arrive, wait until they are called and then are served at some counter until they finally leave the system. Literature can be found in Adan and Resing (2015). All this models and derivations belong to *Bottom-Up processes*, i. e. one starts with model assumptions and calculates the resulting distributions.

E.1 Queueing System of the Basic Model

The basic model can be easily translated in such a queuing process: Customers arrive constantly with rate r_{tran}. We suppose to have infinitely many counters, so no costumer has to wait and is immediately served at one of the counters. Still they need some service time until they are finished and so they leave with rate r_{deg} (see Figure E.1). Adan and Resing (2015) uses the following notation:

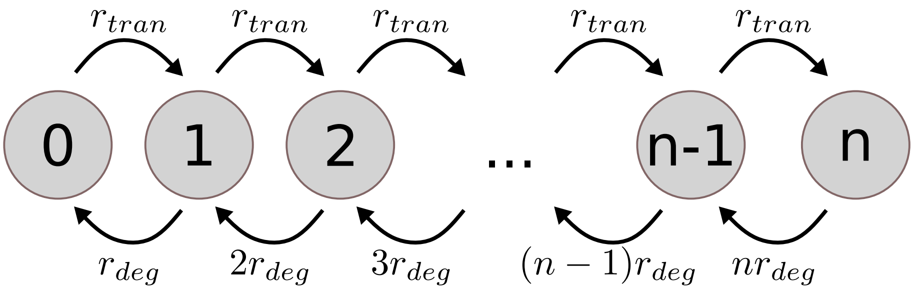


Figure E.1: Basic model is a simple birth-death process. Representation as basic $M/M/\infty$ queue.

- **Distribution of L:** L denotes the number of customers in the system. Denote by p_n the probability that n costumers are in the system.
- **λ:** Arrival rate of the customers

- ρ**:** mean amount of people (= work) that arrives per unit time.
- $\mathrm{E}[B]$**:** mean service time, needed for each customer.

The basic model corresponds to the $M/M/\infty$ queue (Example 11.1.1) which means, that the arriving times are exponentially distributed, the service times are exponentially distributed and infinitely many counter are available. Hence it follows that $\mathrm{E}[L] = \rho$ and at the same time $\rho = \lambda \mathrm{E}[B]$. This means in the settings of the basic model that $\lambda := r_{tran}$ and $B \sim \mathrm{EXP}(r_{deg})$ and hence $\mathrm{E}[B] = \frac{1}{r_{deg}}$. Taken together, this results to

$$\rho = \frac{r_{tran}}{r_{deg}} \tag{E.1}$$

The distribution of L is calculating by equating all flows from state $n-1$ to n and n to $n-1$ (see Figure E.1):

$$\begin{aligned} p_{n-1} r_{tran} &= p_n n \, r_{deg} \\ p_n &= p_{n-1} \frac{r_{tran}}{n \, r_{deg}} \overset{\text{(E.1)}}{=} \frac{\rho}{n} p_{n-1} = \frac{\rho^2}{n(n-1)} p_{n-2} = \ldots = \frac{\rho^n}{n!} p_0. \\ \text{With } 1 &= \sum_{n=0}^{\infty} p_n = p_0 \sum_{n=0}^{\infty} \frac{\rho^n}{n!} = p_0 e^{\rho}, \text{ it follows:} \\ p_n &= \frac{\rho^n}{n!} e^{-\rho}. \end{aligned}$$

This means, that the number of costumers beeing in the system follows a Poisson distribution with parameter $\rho = \frac{r_{tran}}{r_{deg}}$ (see Definition A.7), which is the same result as for the chemical master equation in the previous section.

F Usage of stochprofML

This section illustrates the usage of the **stochprofML** package for simulation and parameter estimation. There are two ways to use the **stochprofML** package: (i) Two interactive functions `stochasticProfilingData()` and `stochasticProfilingML()` provide low-level access to synthetic data generation and maximum likelihood parameter estimation without requiring advanced programming knowledge. They guide the user through entering the relevant input parameters: Working as question-answer functions, they ask for prompting the data (or file name), the number of cells per sample, the number of genes etc. (ii) The direct usage of the package's **R** functions allows more flexibility and is illustrated in the following.

F.1 Synthetic data generation

We first generate a dataset of $k = 1000$ sample observations, where each sample consists of $n = 10$ cells. We choose a single-cell model with two populations, both of lognormal type, i.e. we use the LN-LN model. Let us assume that the overall population of interest is a mixture of 62% of population 1 and 38% of population 2, i.e. $p_1 = 0.62$. As population parameters we choose $\mu_1 = 0.47$, $\mu_2 = -0.87$ and $\sigma = 0.03$. Synthetic gene expression data for one gene is generated as follows:

```
R> library("stochprofML")
R> set.seed(10)
R> k <- 1000
R> n <- 10
R> TY <- 2
R> p <- c(0.62, 0.38)
R> mu <- c(0.47, -0.87)
R> sigma <- 0.03
R> gene_LNLN <- r.sum.of.mixtures.LNLN(k = k, n = n, p.vector = p,
+    mu.vector = mu, sigma.vector = rep(sigma, TY))
```

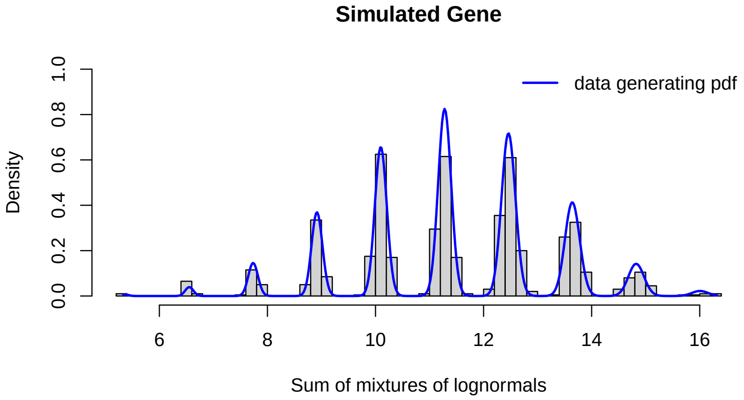


Figure F.1: Histogram of 1000 synthetic 10-cell observations, together with theoretical PDF. We assumed a two-population LN-LN model with parameters $p = 0.62$, $\mu_1 = 0.47$, $\mu_2 = -0.87$ and $\sigma = 0.03$.

Figure F.1 shows a histogram of the simulated data as well as the theoretical PDF of the 10-cell mixture. The following code produces this figure:

```
R> x <- seq(from = min(gene_LNLN), to = max(gene_LNLN), length = 500)
R> stochprofML:::set.model.functions("LN-LN")
R> y <- d.sum.of.mixtures(x, n, p, mu,rep(sigma,TY), logdens = FALSE)
R> hist(gene_LNLN, main = paste("Simulated Gene"), breaks = 50,
+    xlab = "Sum of mixtures of lognormals", ylab = "Density",
+    freq = FALSE, col = "lightgrey")
R> lines(x, y, col="blue", lwd = 2)
R> legend("topright", legend = "data generating pdf", col = "blue",
+    lwd = 2, bty = "n")
```

F.1.0.1 Parameter estimation

Next, we show how the parameters used above can be back-inferred from the generated dataset using maximum likelihood estimation.

```
R> set.seed(20)
R> result <- stochprof.loop(model = "LN-LN",
+    dataset = matrix(gene_LNLN, ncol = 1), n = n, TY = TY,
+    genenames = "SimGene", fix.mu = FALSE, loops = 10,
+    until.convergence = FALSE, print.output = FALSE, show.plots = TRUE,
+    plot.title = "Simulated Gene", use.constraints = FALSE)
```

When the fitting is done, pressing ¡enter¿ causes R to show plots of the estimation process, see Figure F.2, and displays the results in the following form.

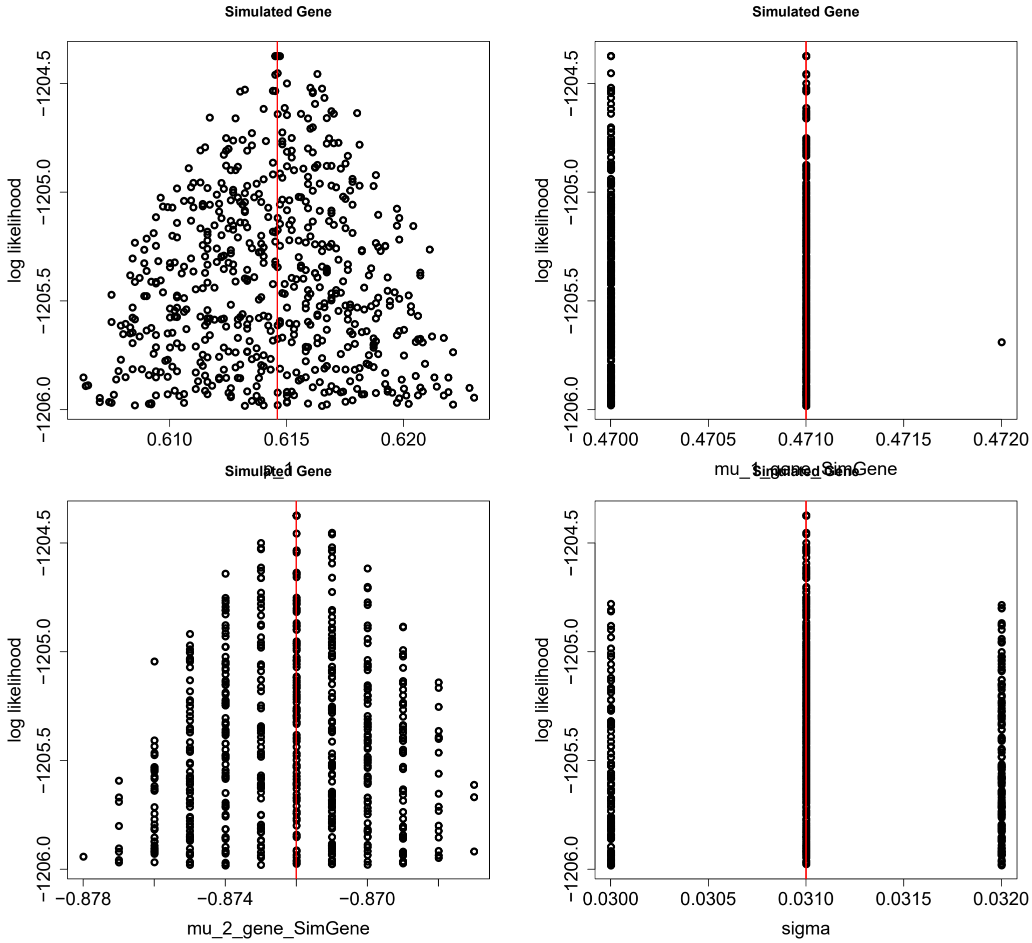


Figure F.2: Graphical output of the parameter estimation procedure for p, μ_1, μ_2 and σ as described in Section Parameter estimation. Each point in the plots corresponds to one combination of values for p, μ_1, μ_2 and σ. Each plot depicts the functional relationship between one parameter (e. g. p in the upper left panel) and the log-likelihood function, whilst the remaining three parameters are integrated out.

```
Maximum likelihood estimate (MLE):
p_1 mu_1_gene_SimGene mu_2_gene_SimGene             sigma
0.6146            0.4710            -0.8720           0.0310

Value of negative log-likelihood function at MLE:
1204.371

Violation of constraints:
none

BIC:
2436.373

Approx. 95% confidence intervals for MLE:
```

```
lower      upper
p_1                 0.60501813  0.6240938
mu_1_gene_SimGene  0.46972264  0.4722774
mu_2_gene_SimGene -0.87827704 -0.8657230
sigma               0.02967451  0.0323847

Top parameter combinations:
p_1 mu_1_ge_SimGene mu_2_gene_SimGene sigma   target
p_1 mu_1_gene_SimGene mu_2_gene_SimGene sigma   target
[1,] 0.6146            0.471            -0.872 0.031 1204.371
[2,] 0.6146            0.470            -0.872 0.031 1204.371
[3,] 0.6146            0.471            -0.872 0.031 1204.371
[4,] 0.6146            0.470            -0.872 0.031 1204.371
[5,] 0.6145            0.471            -0.872 0.031 1204.371
[6,] 0.6146            0.471            -0.872 0.031 1204.371
```

Hence, the marginal confidence intervals cover the true parameter values.

G Further Plots of the stochprofML Simulation Studies

G.1 Impact of pool sizes

In the first simulation study in Section 5.4.1, we investigate how parameter estimation is influenced by increasing cell numbers within the cell pools. The results for parameter set 1 are depicted in the main part of the thesis. Here, we show the corresponding figures for the remaining four parameter settings.

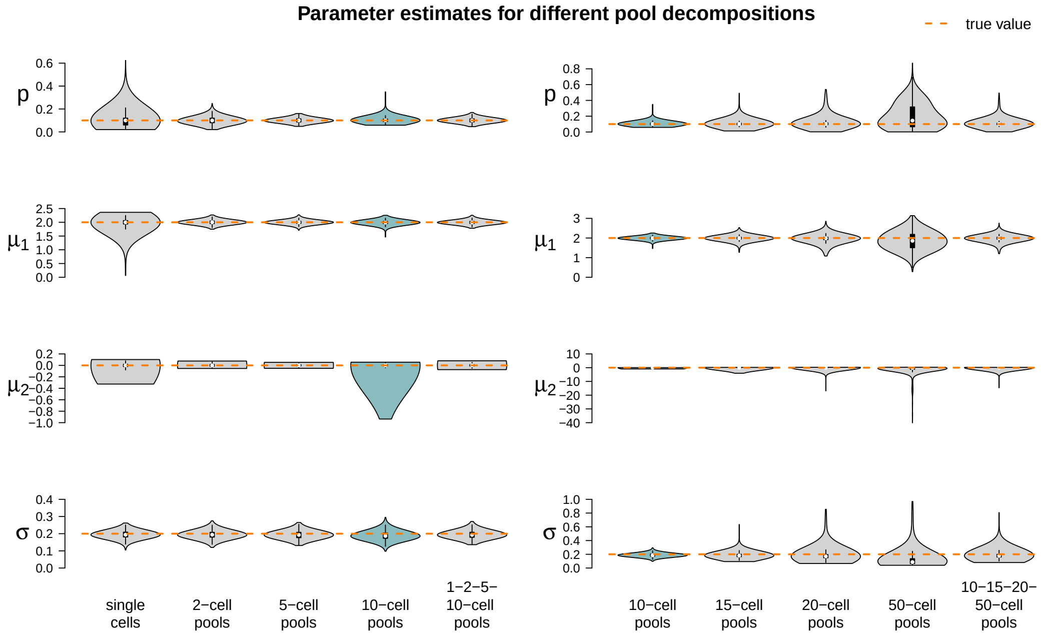


Figure G.1: Estimated parameters of LN-LN-model on 9,000 simulated datasets, i. e. 1,000 datasets of each pool composition generated with parameter set 2 (see Table 5.2). *Left:* Accumulated parameter fits of the single-cell, 2-cell, 5-cell, 10-cell and mixture of single-, 2-, 5- and 10-cell pools. *Right:* Results of the 10-cell pools are repeated (turquoise violins), next to those of the larger pool sizes, namely 15-, 20-, 50-cells and their mixture. Each violin is based on 1,000 parameter estimates. The true parameter values are marked in orange.

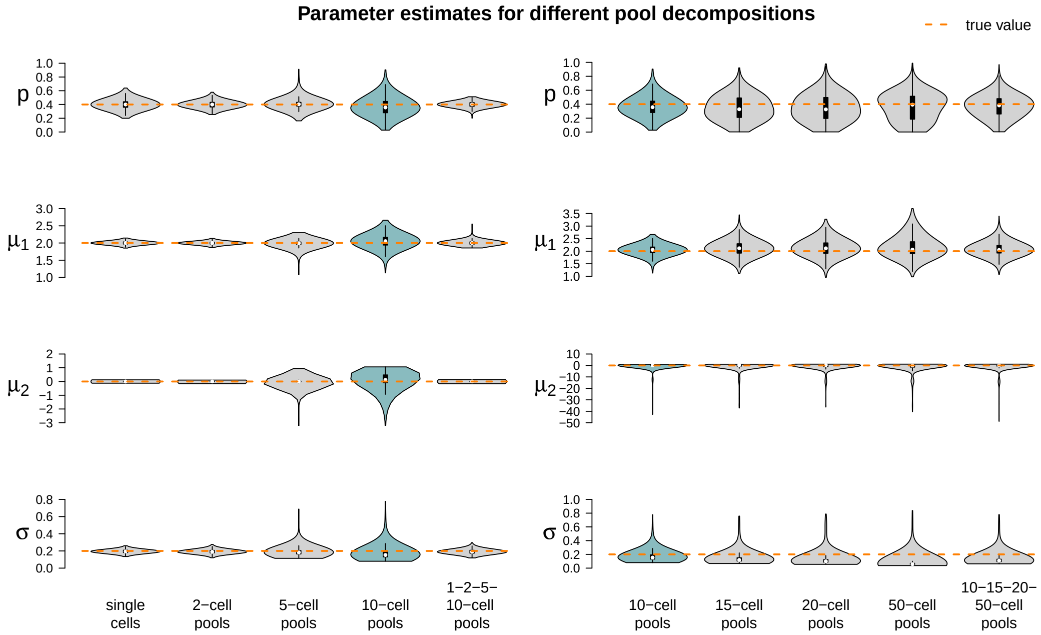


Figure G.2: Parameter estimates as in Figure G.2 but for parameter set 3 (see Table 5.2).

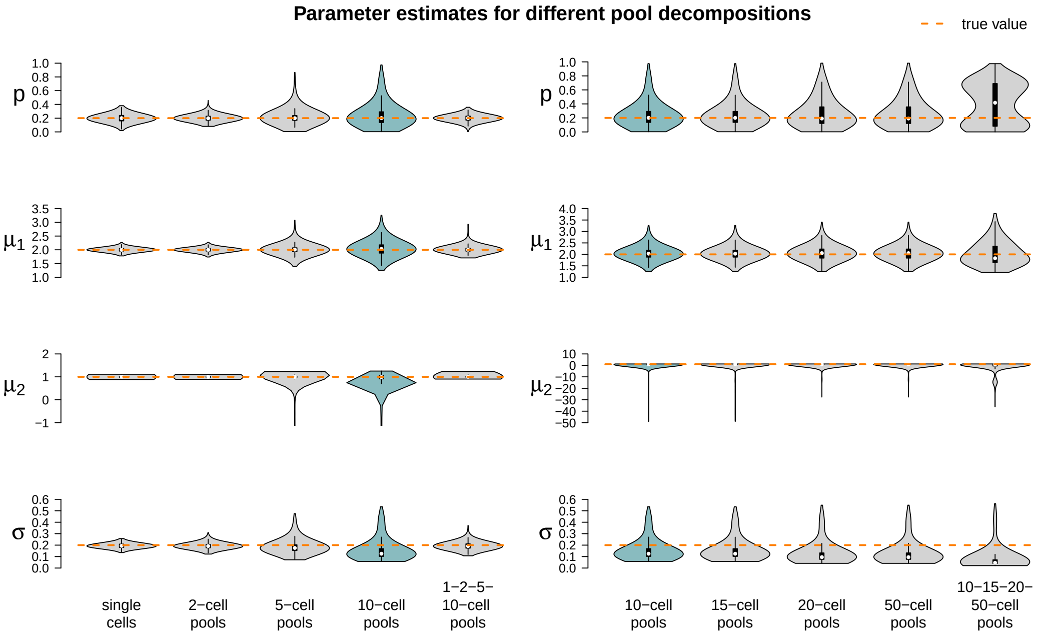


Figure G.3: Parameter estimates as in Figure G.2 but for parameter set 4 (see Table 5.2).

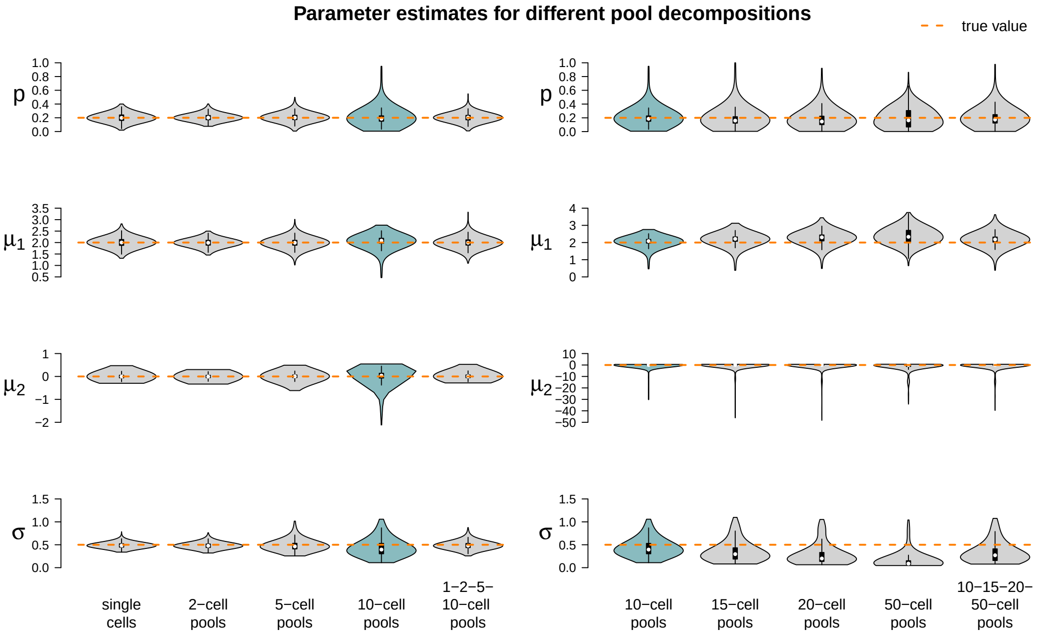


Figure G.4: Parameter estimates as in Figure G.2 but for parameter set 5 (see Table 5.2).

G.2 Impact of parameter values

In Section 5.4.2, we investigate the influence of the model parameter values on the estimation performance while fixing the pool size. In the main part of the thesis, we presented results for 10-cell pools (see Figure 5.5). Here, corresponding figures

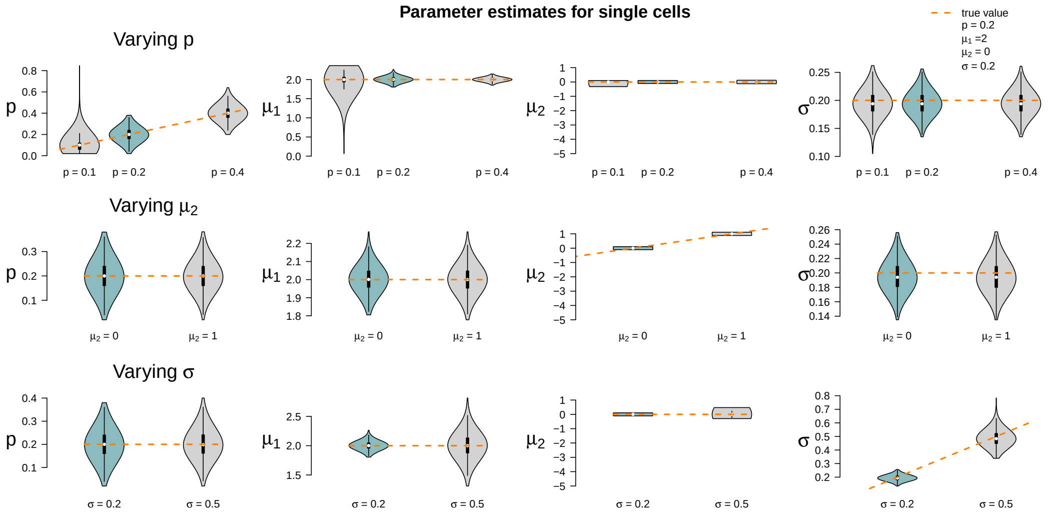


Figure G.5: Parameter estimates for single-cell data and varying parameter values: Synthetic data is generated using the LN-LN model for varying values of p, μ_2 and σ. Results for the standard setting $p = 0.2$, $\mu_1 = 2$, $\mu_2 = 0$ and $\sigma = 0.2$ are shown in turquoise, results for four more settings in grey. For each setting, we generate 1,000 synthetic datasets and back-infer the model parameters. Violin plots summarize the 1,000 estimates. The underlying true parameter values are marked in orange.

for the remaining eight cell pool sizes ($n \in \{1, 2, 5, 15, 20, 50\}$ and two mixtures) are shown.

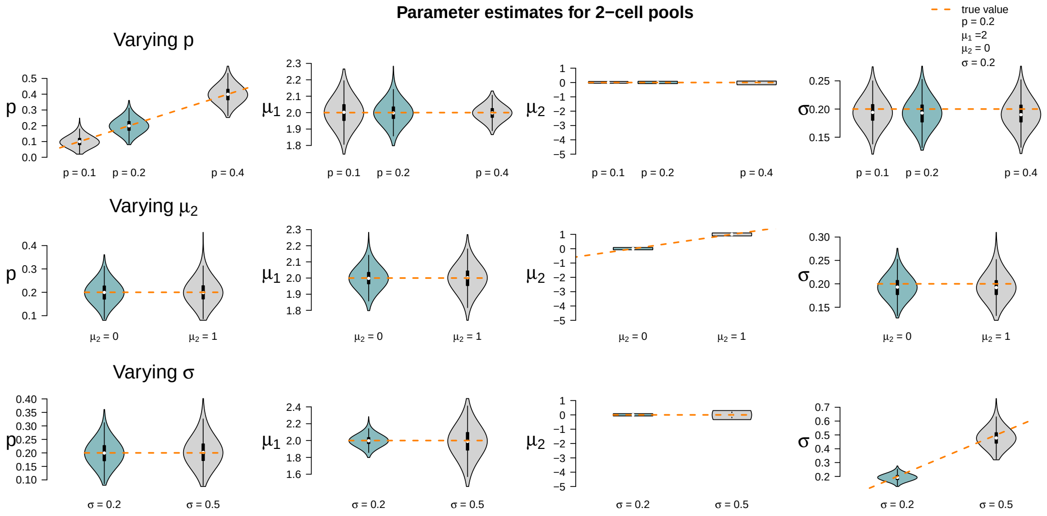


Figure G.6: As Figure G.5, but for 2-cell data.

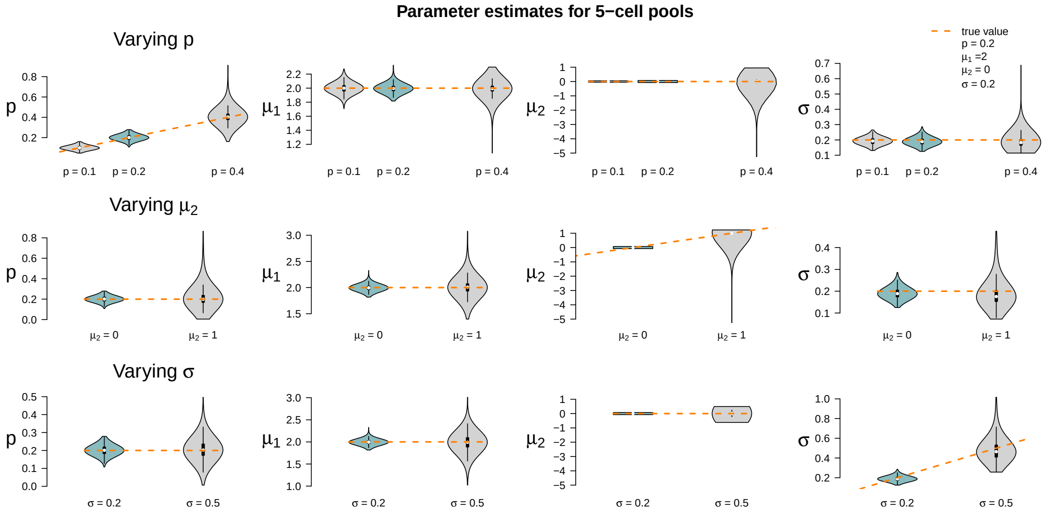


Figure G.7: As Figure G.5, but for 5-cell data.

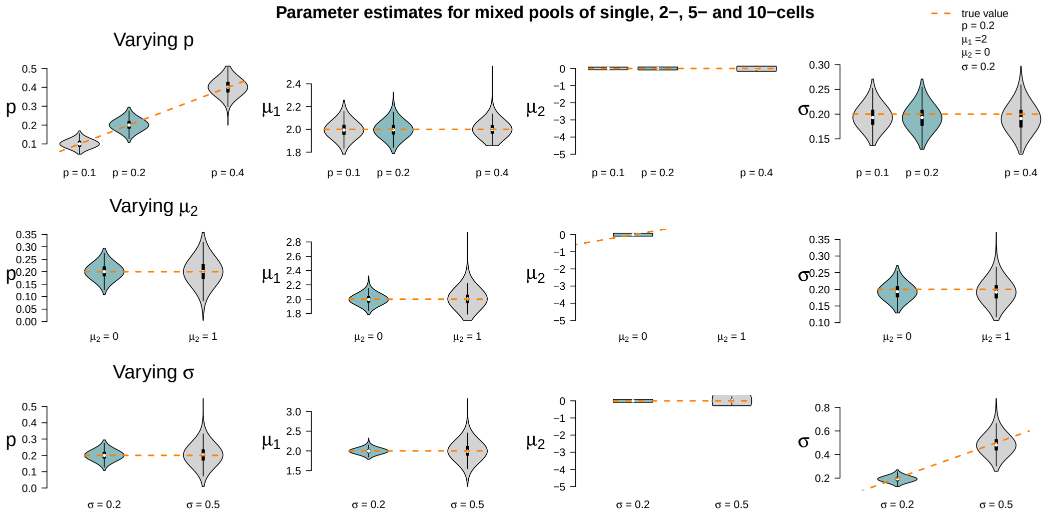


Figure G.8: As Figure G.5, but for a mixture of single-, 2-, 5- and 10-cell data.

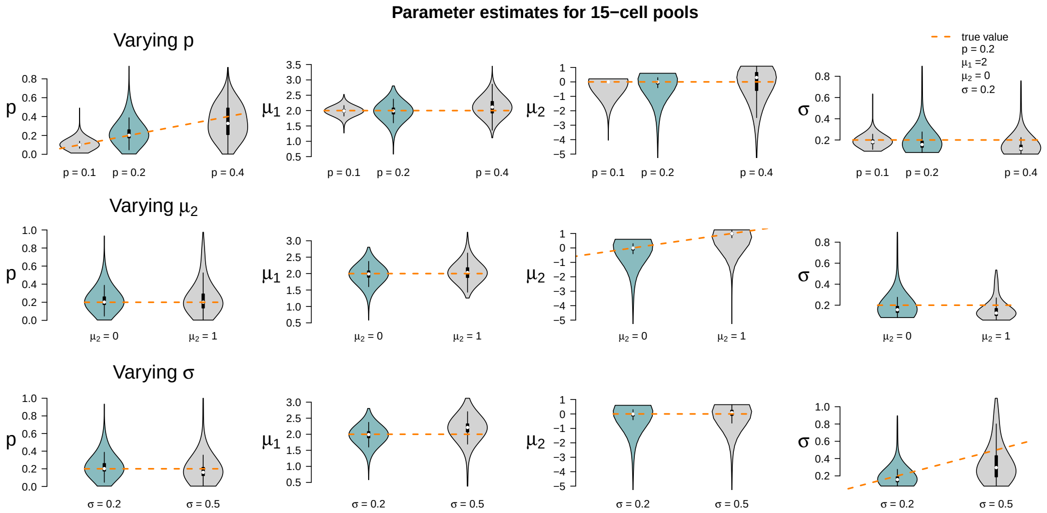
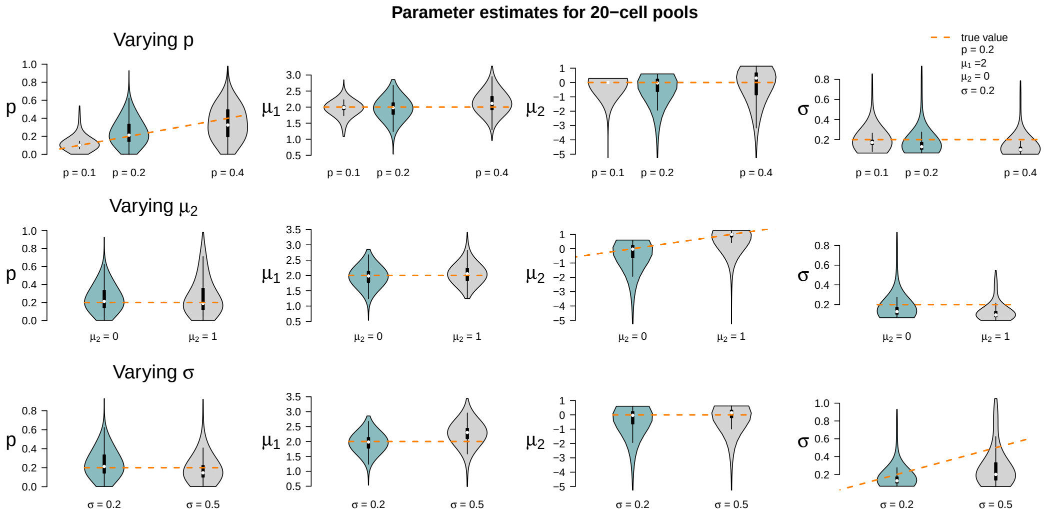


Figure G.9: As Figure G.5, but for 15-cell data.

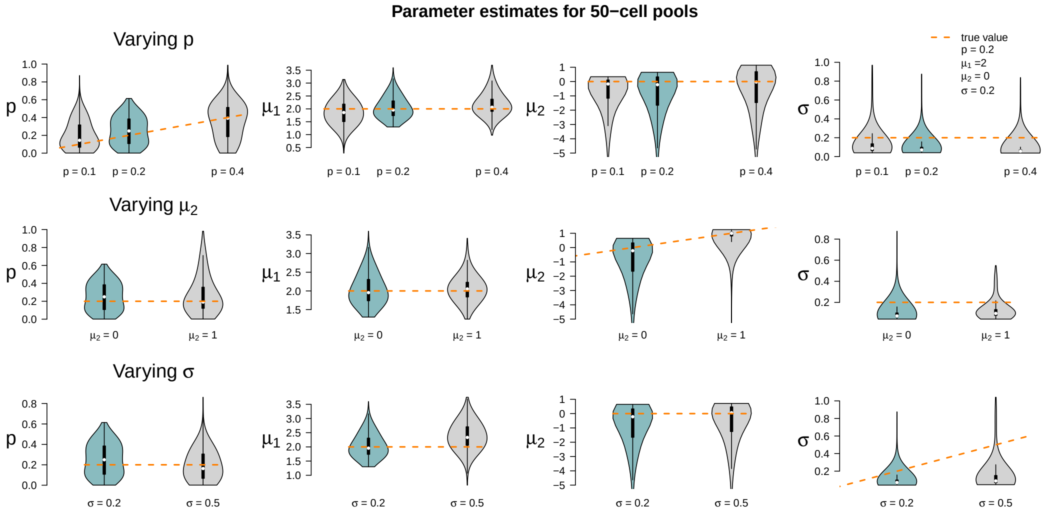

Parameter estimates for 20−cell pools

Varying p

Varying μ_2

Varying σ

Figure G.10: As Figure G.5, but for 20-cell data.


Parameter estimates for 50−cell pools

Varying p

Varying μ_2

Varying σ

Figure G.11: As Figure G.5, but for 50-cell data.

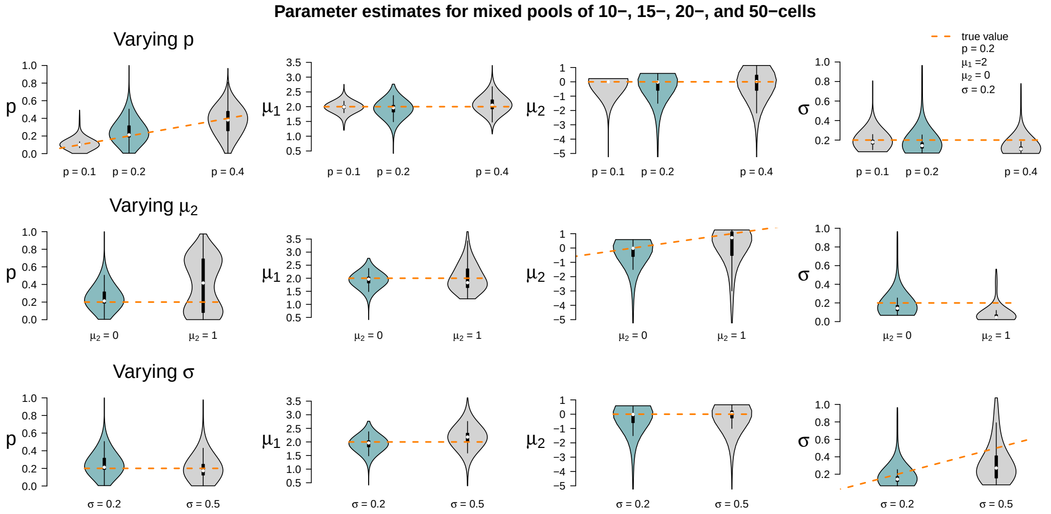


Figure G.12: As Figure G.5, but for a mixture of 10-, 15-, 20- and 50-cell data.

H Details on Stan fits

Here we depict the results of the nine Stan fits used in Section 5.5.3 when comparing them to the corresponding stochprofML runs using the NB-NB model.

1 Population

Single-cells

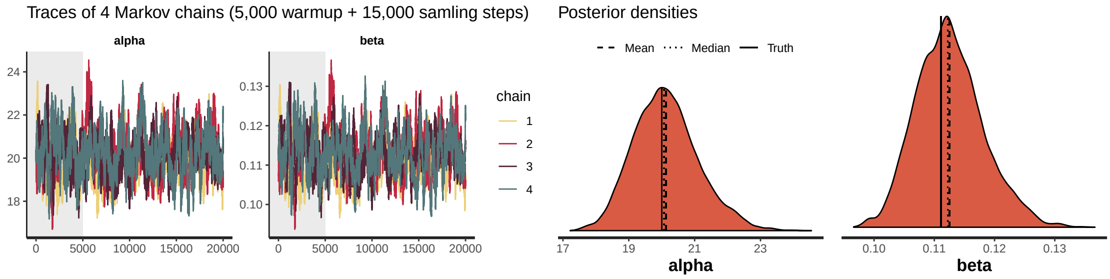


Figure H.1: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on single-cell data fitting one population.

Pool of 2 cells

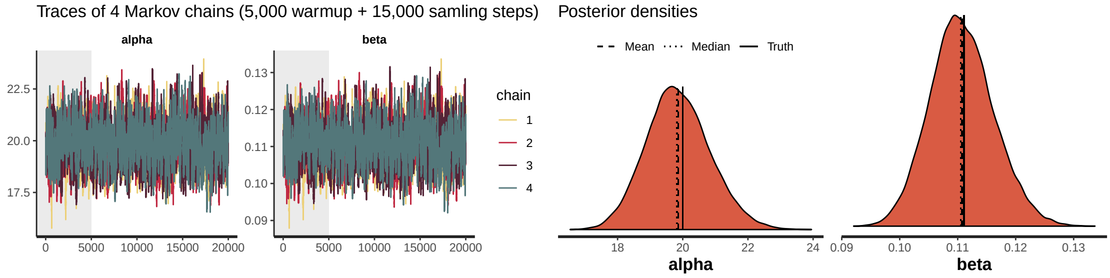


Figure H.2: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 2-cell data fitting one population.

Pool of 5 cells

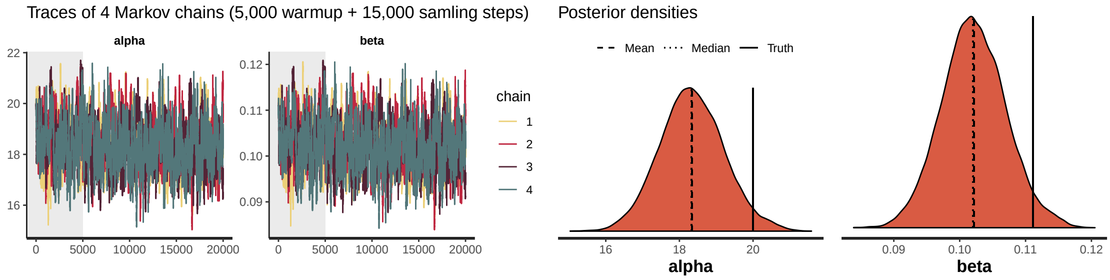


Figure H.3: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 5-cell data fitting one population.

Pool of 10 cells

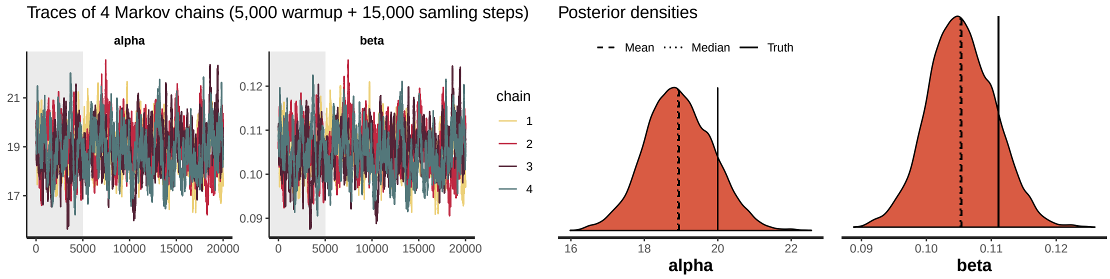


Figure H.4: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 10-cell data fitting one population.

2 Populations

Single-cells

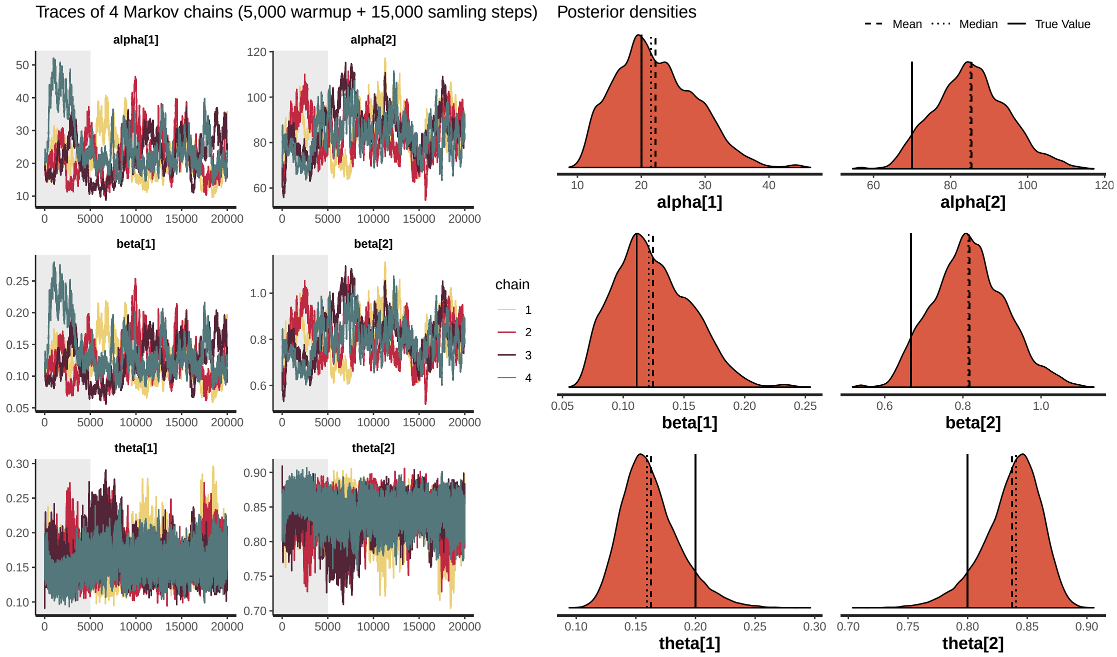


Figure H.5: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on single-cell data fitting two populations.

Pool of 2 cells

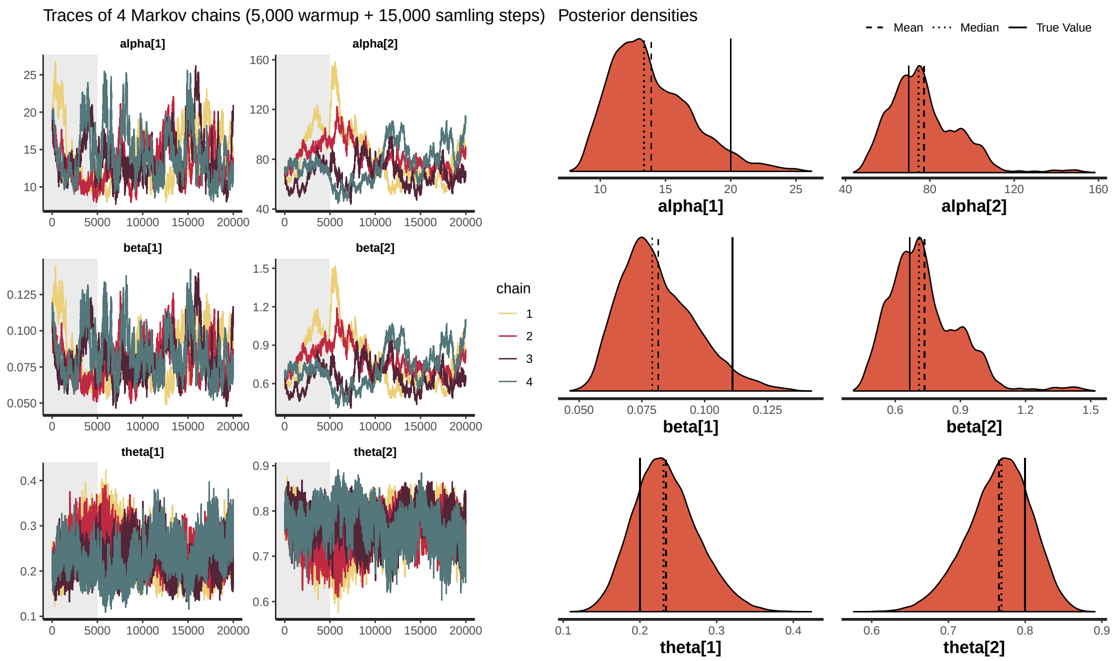


Figure H.6: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 2-cell data fitting two populations.

Pool of 5 cells

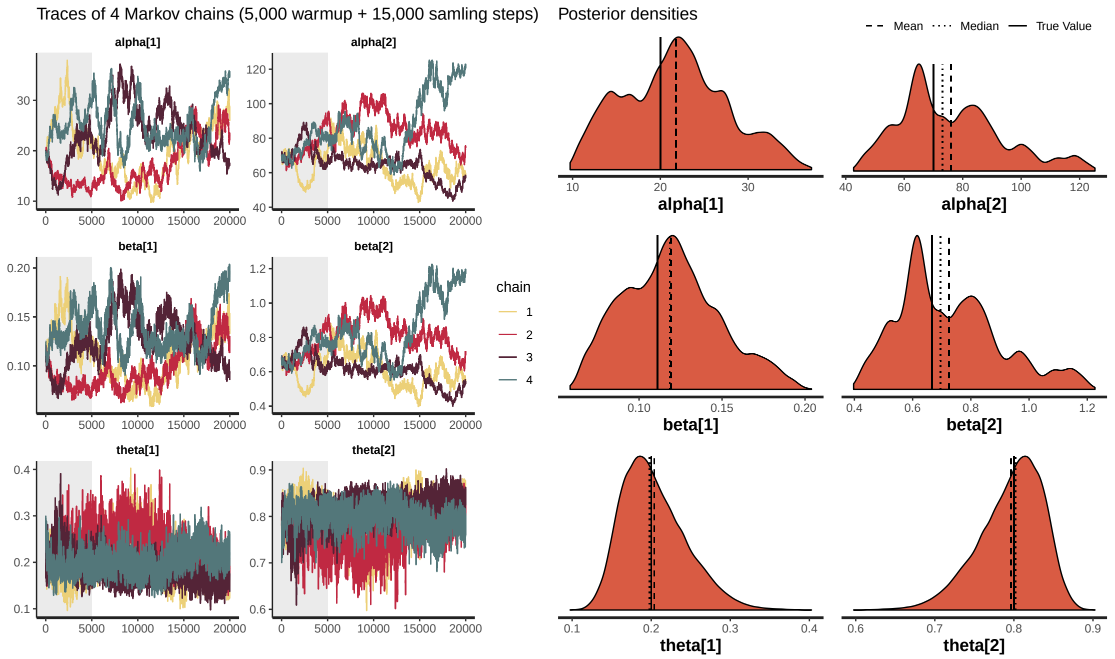


Figure H.7: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 5-cell data fitting two populations.

Pool of 10 cells

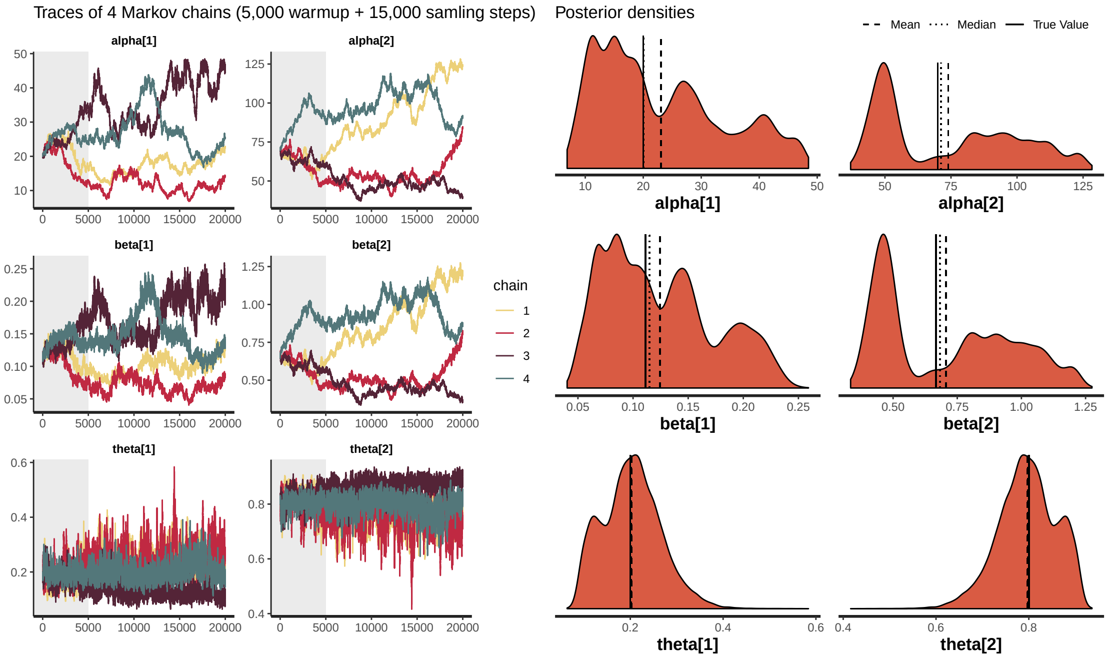


Figure H.8: Parameter traces and densities of the posterior sample obtained with the NUTS using the PG implementation on 10-cell data fitting two populations.

I Additional Plots and Results of SFB Data

I.1 Descriptive Plots and GAMLSS Results

Figure I.1 shows the UMAPs before and after ComBat-seq of the mESC dataset. Figure I.2 contains the AML data.

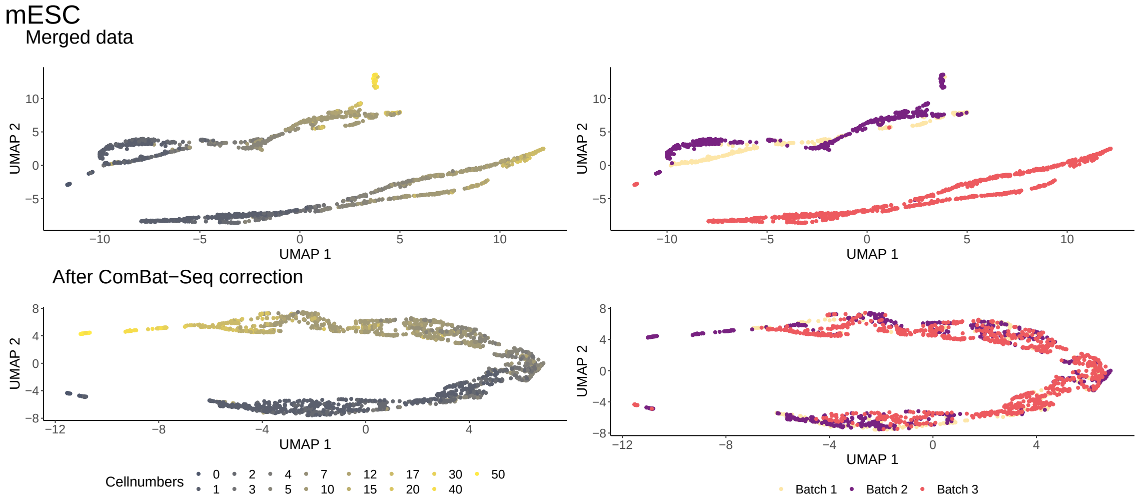


Figure I.1: UMAPs of the UMIs of the merged mESC datasets. The bottom row shows the UMAP after batch correction via ComBat-seq. Colors identify cellnumbers (left) and the three experimental batches (right).

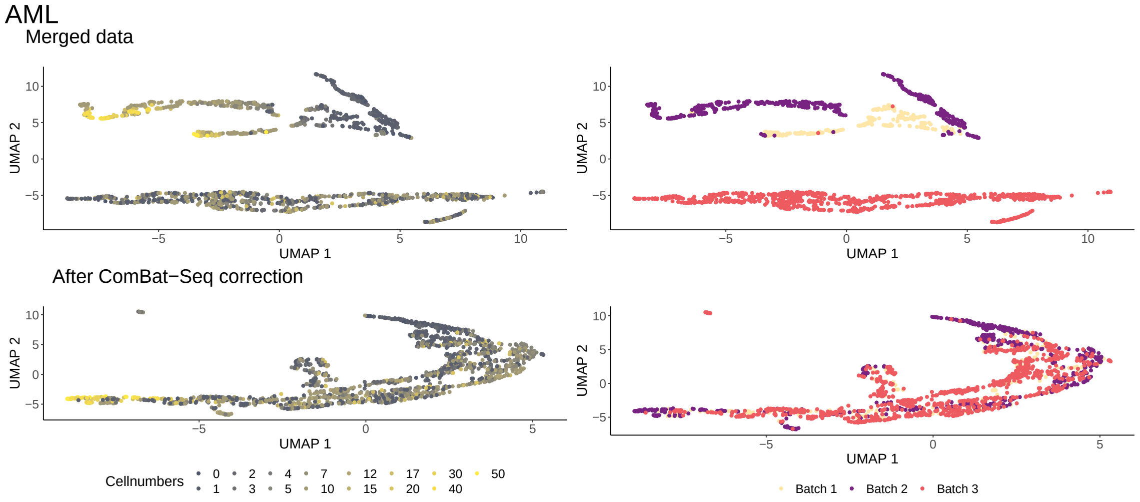


Figure I.2: UMAPs of the UMIs of the merged AML datasets. The bottom row shows the UMAP after batch correction via ComBat-seq. Colors identify cellnumbers (left) and the three experimental batches (right).

A GAMLSS negative binomial regression with cellnumbers as covariates and dates as mixed effects is shown in Figure I.3. A GAMLSS after batch correction where no mixed effectes are included is shown in the main text in Figure 6.6.

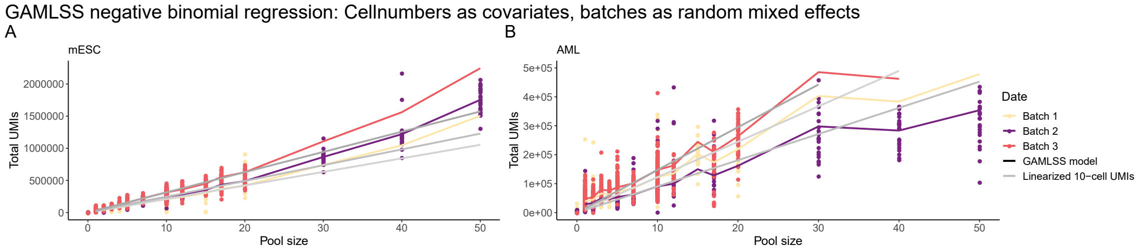


Figure I.3: GAMLSS NB regression model using the non corrected UMI datasets of the mESC data (A) and the AML data (B). Cellnumbers serve as covariates and batches as mixed effects. For comparison the induced linear relationship by the 10-cell UMI content is added in grey for each batch.

A comparison of mean single-cell UMI counts per cell for different pool sizes are shown in Figure I.4. Figure 6.7 in the main part of this thesis zoomed into this figure.

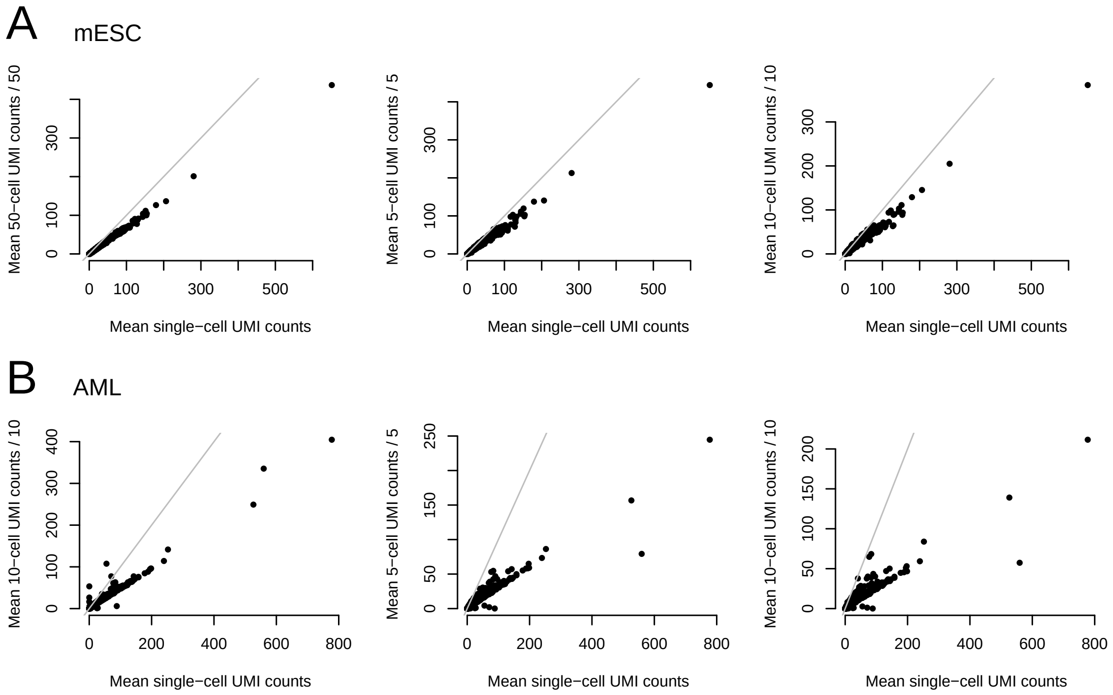


Figure I.4: Mean single-cell UMI counts per gene compared to the normalized mean UMI-count per cell for 2-cell pools, 5-cell pools and 10-cell pools. The mESC dataset is depicted in A and the AML dataset in B. The gray line describes the case that both means are the same.