Computational studies of genome evolution and regulation

Karina Zile

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Institute of Structural and Molecular Biology (ISMB)

University College London

March 3, 2020

I, Karina Zile, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis takes on the challenge of extracting information from large volumes of biological data produced with newly established experimental techniques. The different types of information present in a particular dataset have been carefully identified to maximise the information gained from the data. This also precludes the attempts to infer the types of information that are not present in the data. In the first part of the thesis I examined the evolutionary origins of *de novo* taxonomically restricted genes (TRGs) in *Drosophila* subgenus. *De novo* TRGs are genes that have originated after the speciation of a particular clade from previously non-coding regions - functional ncRNA, within introns or alternative frames of older protein-coding genes, or from intergenic sequences. TRGs are clade-specific toolkits that are likely to contain proteins with yet undocumented functions and new protein folds that are yet to be discovered. One of the main challenges in studying *de novo* TRGs is the trade-off between false positives (non-functional open reading frames) and false negatives (true TRGs that have properties distinct from well established genes). Here I identified two *de novo* TRG families in *Drosophila* subgenus that have not been previously reported as *de novo* originated genes, and to our knowledge they are the best candidates identified so far for experimental stud-

ies aimed at elucidating the properties of *de novo* genes. In the second part of the thesis I examined the information contained in single cell RNA sequencing (scRNA-seq) data and propose a method for extracting biological knowledge from this data using generative neural networks. The main challenge is the noisiness of scRNA-seq data - the number of transcripts sequenced is not proportional to the number of mRNAs present in the cell. I used an autoencoder to reduce the dimensionality of the data without making untestable assumptions about the data. This embedding into lower dimensional space alongside the features learned by an autoencoder contains information about the cell populations, differentiation trajectories and the regulatory relationships between the genes. Unlike most methods currently used, an autoencoder does not assume that these regulatory relationships are the same in all cells in the data set. The main advantages of our approach is that it makes minimal assumptions about the data, it is robust to noise and it is possible to assess its performance. In the final part of the thesis I summarise lessons learnt from analysing various types of biological data and make suggestions for the future direction of similar computational studies.

Impact Statement

Single cell RNA sequencing (scRNA-seq) is a recently developed experimental technique that captures the transcriptional heterogeneity between individual cells. The data produced enables biological discoveries that were not previously possible due to the lack of resolution. Projects based on scRNA-seq data result in important contributions in developmental and regeneration biology, and in studies of the immune system and ageing. In 2019 alone numerous landmark papers have been published reporting a diverse set of discoveries. For example, it has been shown that different cell types age in unique ways and mechanisms of disrupted cardiac development have been identified. Cell lineages and gene networks for neural subtypes constituting the whole larval nervous system of a sea squirt have been inferred. The Human Liver Cell Atlas and the Mouse Organogenesis Cell Atlas providing a global view of developmental processes in mice have been published, and the work on the Human Cell Atlas is ongoing. Motivated by these outstanding results, substantial resources are being allocated to producing more scRNA-seq data. The computational analysis of this type of data remains challenging due to its compositional nature and due to the complex structure of noise contained therein. Significant resources are being dedicated to developing new analysis methods, but due to the lack

of ground truth it is challenging to assess their performance. Benchmarking studies report inconsistent results for groups of similar methods. Existing analysis methods are based on strong assumptions about the data. Here I examine these assumptions in the context of mathematical properties of scRNA-seq data, thus offering an explanation as to the poor results observed in benchmarking studies. I propose an analysis method based on generative neural networks. This method makes minimal assumptions about the data and is able to integrate information from datasets produced by different labs. The method is scalable and flexible - it can be easily applied to large amounts of data, and executed either on GPU clusters or conventional desktops with limited RAM via data streaming. I also examine different approaches for modelling the information flow in a cell in a more biologically meaningful way, actively shaping the training process of neural networks to achieve desired properties, and training the models in a reproducible manner. My work will enable better outcomes of future studies based on scRNA-seq data. It will contribute to the adoption of clinical applications of this type of data, which is currently inhibited by the lack of reproducible and assessable analysis methods. The impact of my work will increase with the maturation of single cell technologies that are able to capture both transcriptomic and other types of data from the same cells, as using machine learning will be the dominant strategy for analysing this new type of data. Both my work on scRNA-seq data analysis and on identifying evolutionary origins of taxonomically restricted gene will become a stepping-stone for future research in these areas.

Acknowledgements

I would like to say a very big thank you to my supervisors Nick Luscombe and Kaila Srai for their infinite faith in me and for always helping me when I needed it most. I am grateful to Joanna Masel for being a great mentor and for volunteering to take on this unofficial role. I greatly appreciate the support from everyone in Luscombe group. A special thank you to Beibei Xinyi Du for saving my life when Varicella zoster virus attacked me just a couple of weeks before submitting this thesis. Another special thank you to Slava Sidorov for teaching me IT-speak in Russian. Thanks to Jeremy Levy and Alex Warwick Vesztrocy for literally always having my back and being there for me in the best and worst of times. I am grateful to Laura Piovani for being a true friend and for always cheering me up, sometimes by showing me her newborn snail twins. A big thank you to Steven Müller for greatly improving my PhD experience with his support and boardgames. Thanks to Emeline Favreau for being a great human and to Rodrigo Pracana for letting me garden in his vegetable patch. Thanks to the Francis Crick Institute Running Club for keeping me mentally and physically fit, and for showing me that I can run 45 kilometres. I deeply appreciate my best friend Johannes Welbl for being my mental compass and for tolerating my middle of the night phone calls. I am especially

grateful to the love of my life Harry Lascelles for supporting me every inch of the way and always reminding me that there are no limits to what I can achieve. This work would not have been possible without the funding provided by Biotechnology and Biological Sciences Research Council [grant number BB/M009513/1].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The expression “genomical amounts of data” is rightfully replacing previously used “astronomical amounts”. In their projection to year 2025, Stephens et al. [2015] showed that genomic data is likely to be the biggest of the Big Data domains. Development of new experimental techniques has increased the amount of biological data generated by many orders of magnitude, and this has resulted in the need of new methods for handling and analysing that data and for interpreting the results. The nature of biological data has changed, while methods are still catching up.

This thesis takes on the challenge of extracting information from large volumes of biological data produced with newly established experimental techniques. The different types of information present in a particular dataset have been carefully identified to maximise the information gained from the data. This also precludes the attempts to infer the types of information that are not present in the data. For example, in the case of inferring evolutionary origins of protein coding genes, both synteny and homology information is used to infer the evolutionary mechanism,

while no inference is made across large evolutionary distances where sequence conservation is insufficient. Similarly, information about highly expressed genes and relative expression levels in single cell RNA-seq data is exploited in neural network training, while gene expression levels are never compared across cells due to the lack of comparable values. I will critically re-evaluate assumptions typically made about specific types of data and propose analysis methods that are not hampered by fundamentally wrong but seemingly necessary simplifying assumptions about the data.

With this shift in the nature of biological data comes both new challenges and many great opportunities. Equally, computer science is experiencing an accelerated pace of discovery and innovation. As UCL's PhD graduate, now CEO of DeepMind, Demis Hassabis said "some of the most interesting areas of science are in the gaps between subjects". The ambition of my work at the interface of biology and computer science is to build on my expertise in both domains, combining cutting edge research from fields across disciplines and advancing our ability to extract information from genomical amounts of biological data.

This thesis contains two parts - a computational study of genome evolution and of genome regulation. In the introduction chapter I will provide a comprehensive review of the relevant fields and the methods commonly employed to answer biological questions that are at the core of these fields of study. I will conclude this chapter by discussing the intended contribution of my work. Chapter 2 contains

the results of my work aimed at pushing the boundaries of what we can learn about protein-coding genes evolving *de novo* using computational approaches. In the discussion part of this chapter I will conclude that further advancement of our knowledge about *de novo* genes requires experimental approaches. Chapter 3 contains the results of my work aimed at analysing single cell RNA-seq (scRNA-seq) data with generative neural networks. This work combines the most recent advances in the two fields. When I started my PhD three years ago, this work was not possible as the data of this type and scale was not yet available and the advancements in machine learning theory employed here have not yet been made. After a thorough assessment of the mathematical underpinnings of scRNA-seq analysis with generative neural networks and an in depth analysis of advantages and limitations of this approach, Chapter 4 provides an exploration of different ways how these advantages could be exploited more fully and how some of the limitations could be avoided. In this chapter I combine knowledge about the properties of scRNA-seq and the flow of information through a living cell with recent theoretical advancements in neural network training. In the discussion chapter I will put my work on scRNA-seq into the context of the ongoing efforts in this field both from the biology and the computer science fronts. After summarising the contributions of my work and discussing how they fit within a wider research agenda I will share my vision about the future of single cell techniques and about the future of machine learning in biological research. This chapter will not discuss my work on evolutionary origins of taxonomically restricted genes, since experimental studies are required to advance our knowledge and this is beyond the scope of this thesis.

1.1 Computational study of genome evolution

The decrease in genome sequencing costs resulted in an establishment of a new field of research focused on evolution of genes and species. New computational methods for species tree inference took over from morphology and taxonomy studies. More genomes being available also created an opportunity to trace the evolutionary origins of gene families, due to the pairs of homologous genes that are separated by a shorter evolutionary distance and hence could now be identified. Increased number of homologs in gene families subsequently allowed for building sequence profiles and employing those to identify more distant homologs. In turn, this accelerated our ability to predict functions of genes computationally, since experimentally established functional data could now be transferred across species, even distantly related ones.

In parallel to gene homology studies, which are focused on identifying evolutionary related genes based on sequence similarity, there emerged the new field of study of recently evolved new genes. The idea of new functional genes evolving from scratch captivated the imagination of researchers. The problem was that sequenced genomes belonged to species separated by large evolutionary distances. At those distance, sequence and synteny conservation was insufficient to distinguish between genes that evolved by divergence from an ancestral gene and genes that evolved from previously non-coding genome regions. With thousands of se-

quencing projects in the pipeline, computational studies of new genes are finally feasible. Instead of speculating whether the sequence of a gene is different enough from all other sequenced genes to believe that this gene has evolved *de novo*, it is now possible to trace the evolutionary origins of genes along the branches of a species tree with closely related sequenced genomes. In this section I will provide an overview of the field of study of new genes that evolved *de novo* from previously non-coding parts of the genome, from the paper that coined the term *orphan* gene to controversial debates that currently divide researchers in this field.

1.1.1 *De novo* evolution of protein-coding genes

Some genes are present only in one clade, and are therefore called taxonomically restricted genes (TRGs). They are also referred to as *orphans* or simply novel genes. Before I can precisely define a taxonomically restricted gene, a definition of a gene is required. Numerous different definitions have been employed throughout history, see Portin and Wilkins [2017] for a comprehensive overview. Here I will adopt the gene definition proposed by Portin and Wilkins [2017]: A gene is a DNA sequence (whose component segments do not necessarily need to be physically contiguous) that specifies one or more sequence-related RNAs/proteins that are both evoked by genetic regulatory networks and participate as elements in genetic regulatory networks, often with indirect effects, or as outputs of genetic regulatory networks, the latter yielding more direct phenotypic effects. This definition highlights how genes are different from other functional elements in the genome,

for example promoters and enhancers. In this work I will only consider protein coding TRGs, even though non-coding genes can also be taxonomically restricted. Some of these protein coding TRGs may have originated *de novo*, i.e. from non-coding regions [Vakirlis et al., 2017, McLysaght and Guerzoni, 2015], *de novo* in alternative frames of established genes [Willis and Masel, 2018, Guan et al., 2018], or as a result of genome rearrangement [Chen et al., 2015, Stewart and Rogers, 2019]. There are different ways in which the concept of origination or gene birth can be defined. Since only protein coding genes will be considered in this work, I chose the following definition that is concrete yet not too restrictive. The birth of a protein-coding gene is associated with the moment beyond which a mutation leading to loss of translation would have a negative effect on fitness. For this to occur, a *de novo* TRG needs not only the amino acid sequence itself, but also the right environment and expression regulation pattern to confer an advantage to the organism.

The research enterprise is biased toward studying ancient gene families with homologs, i.e. genes evolved from a common ancestor gene, across multiple model organisms. The properties and evolutionary dynamics of young TRGs are not well understood, since they are present in at most one model organism and are often expressed only in a specific tissue or a specific developmental stage. TRGs are likely to include proteins with yet undocumented functions and, especially in the case of *de novo* genes, new protein domains or other structural forms that are yet to be discovered [Bungard et al., 2017]. Mounting evidence suggests that TRGs can acquire important functions. For example, a TRG in the tardigrade *Ramazzottius*

varieornatus produces a protein that protects DNA and improves radio-tolerance [Hashimoto et al., 2016]. TRGs in *Hymenoptera* are implicated in the speciation of parasitoid wasps and in the production of diverse venoms characteristic of this clade [Werren et al., 2010]. Albertin et al. [2015] identified numerous cephalopod-specific genes and were able to find hints about their diverse functions based on their tissue specific expression profiles. These examples remain anecdotal since neither structural nor functional characteristics of TRGs can be inferred computationally due to the lack of homologs outside a specific clade.

Just several decades ago it was believed that TRGs simply do not exist. Susumu Ohno was convinced that "each new gene must have arisen from an already existing gene" and "in a strict sense, nothing in evolution is generated *de novo*" [Soukup, 1974]. In an influential essay, Francois Jacob strengthened this view by stating that "the probability that a functional protein would appear *de novo* by random association of amino acids is practically zero" [Jacob, 1977]. In the 1970s, gene duplication was considered to be a single most important mechanisms in creating new genes by divergence from an ancestral gene [Ohno, 1973]. While gene duplication is still believed to be the main source of new genes, there is now an increased interest in other ways the genes can originate. The term orphan was first defined by Dujon [1996] in the section called "The mystery of orphans" which focused on the large proportion of yeast genes that had no known homologs. In the same issue of Trends in Genetics Casari et al. [1996] predicted that with influx of new sequencing data the number of *orphans* will soon reach zero, especially given that their number

was already down to 10%. As recently as 2003, TRGs were still thought to be simply an artefact of limited number of sequenced genomes [Siew and Fischer, 2003]. The molecular details of the origin of new genes have been rigorously examined for the first time by Long et al. [2003]. The work by [Begun et al., 2005] resulted in the first published claim that genes could potentially originate *de novo*. Two decades and thousands of sequenced genomes later, the prediction made by Casari et al. [1996] remains unfulfilled - the proportion of TRGs in newly sequenced genomes is still around 10%.

Now there is mounting evidence that TRGs do emerge *de novo*, but both the definition of *de novo* TRGs and what it means for a gene to emerge remain highly controversial topics [Schlötterer, 2015]. For example, a gene whose sequence diverged from an ancestral gene sequence beyond detectability is indistinguishable from a *de novo* TRG. Similarly, a gene acquired through a horizontal gene transfer from a distant clade can be easily mistaken for a *de novo* gene. In both cases, the gene is likely to perform a different function in the new genomic context or compared to the ancestral gene. Similarly, a gene that is conserved only in a specific clade and whose homologs have been lost in every other lineage is indistinguishable from a *de novo* TRG [Morel et al., 2015]. As a result, there is no clear line and the definition of *de novo* TRGs remains inconsistent across literature.

In the case of a true *de novo* gene, the birth of protein-coding gene is associated with the moment beyond which a mutation leading to loss of translation would

have a negative effect on fitness. From that moment onwards this transcribed and translated sequence becomes visible to selection; and it is also more likely to be conserved and hence to have more opportunities to evolve. Before that the sequence is neither conserved nor selected for its protein-coding properties, and hence cannot be regarded as a protein-coding gene. This moment of gene birth must be associated with one of the four events: a transcript acquires an ORF, an ORF becomes transcribed, a transcribed ORF becomes beneficial to the organism due to a change in it's environment, a transcribed ORF acquires a mutation. Here mutation is meant in a broad sense, it could be a change in ORF's amino acid sequence or an altered expression profile - expression up-regulation, expression in a different tissue or at a different developmental stage. A lot of work has been dedicated to gathering evidence for two competing hypotheses - whether the *de novo* gene emerge ORF-first or transcript-first, see Schlötterer [2015] for a review of this debate. Neither a non-transcribed ORF nor a transcript without an ORF can be a functional protein-coding gene, and hence neither of them are under conservation/selection for their properties related to being a protein-coding gene. Another line of debate is concerned with whether the transition from a non-coding region to a functional protein-coding gene could happen through an intermediary stage [Carvunis et al., 2012]. The work by Wilson et al. [2017] provides evidence against this continuum hypothesis. The role of the stop codons in the process of *de novo* gene emergence also remains controversial. Short ORFs are less likely to be prone to aggregation and hence they are believed to be the main source of *de novo* gene [Wilson et al., 2017]. At the same time, probability of acquiring an in-frame stop codon anywhere

along the ORF is much higher than the probability of getting rid of a particular stop codon that would result in an extension of the ORF. The exact ratio between these probabilities depends on the GC content of a particular sequence, but in any conceivable protein-coding sequence the probability of getting rid of a particular stop codon (i.e. of a mutation in one of those three nucleotides or a frameshift bringing the stop codon out of frame) is always lower than the combined probability of any codon in the sequence mutating into a stop codon and any pre-existing out-of-frame stop codon shifting into frame due to an indel mutation. The roll of translational stop codon read-through is debated both in the context of emergence of longer ORFs [Jungreis et al., 2016] and in the context of sequence pre-adaptation that facilitates future *de novo* gene emergence [Wilson et al., 2017]. Pre-adaptation hypothesis argues that harmful ORFs disappear rapidly from the pool of sequences available for transcription aided by pervasive transcription and pervasive translation [Ruiz-Orera et al., 2018]. Jungreis et al. [2016] showed that translational stop codon read-through are under continued purifying evolutionary selection in *A. gambiae* mosquito and that they are sometimes associated with new gene birth in both *A. gambiae* and *D. melanogaster*.

There are numerous *de novo* gene related open questions that are currently under active investigation. How often do *de novo* genes emerge and what properties of the genome does this emergence rate depend on? Are *de novo* genes lost more often than well established genes? How do they integrate into gene regulatory network? Is phenotypic novelty, for example a new type of tissue, a new developmental stage

or a new venom/pheromone produced, more often associated with *de novo* genes than not? Are *de novo* genes associated with new, previously unobserved types of protein folds? Answering most, if not all, of these questions requires a decently sized sample of *de novo* genes which could then be compared to a sample of well established genes. Hence, the ability to identify *de novo* genes in different clades and to distinguish them from evolved copies of well established genes is essential to advancing our understanding of evolutionary dynamics of *de novo* genes.

1.1.2 Identifying taxonomically restricted genes

The first challenge in studying *de novo* TRGs is identifying them. Most previous studies aimed at elucidating properties and rates of emergence of *de novo* TRGs have used an approach known as "phylostratigraphy" that focuses on protein-coding genes with protein homologs within a particular clade and no detectable homology outside that specific clade. This approach is incapable of discriminating between *de novo* genes and highly diverged copies of well-established genes. Hence, the properties of "young genes" reported in these studies are averages computed across the two groups, and risk attributing to TRGs properties that instead reflect the disappearance of the ability to detect homology. For example, most of the studies reported that new genes tend to be shorter [Wissler et al., 2013, Zhao et al., 2014, Ruiz-Orera et al., 2015, Sun et al., 2015] and evolve faster than well established genes [Domazet-Loso, 2003, Toll-Riera et al., 2008, Donoghue et al., 2011]. It is palusable that TRGs should be shorter than well established genes, since short

ORFs are less likely to be prone to aggregation [Wilson et al., 2017]. It is also plausible that TRGs could evolve faster than well established genes, because there are less evolutionary constraints associated with them. They are less likely to have many interaction partners and to be entangled in a gene regulatory network. Hence, it is *a priori* plausible that TRGs have these properties, but phylostratigraphy does not provide clear evidence to support this claim because these properties are also associated with diminishing ability to infer homologous sequences. It is harder to detect homology for shorter and/or faster evolving genes, and this is sufficient to explain at least the qualitative direction of the observed trend. Arendsee et al. [2019] showed that including synteny information in the phylostratigraphy analysis changes the inferred gene ages, thus demonstrating that by itself phylostratigraphy approach is not sufficient.

There is an ongoing debate about the frequency of *de novo* gene birth [Casola, 2018]. While the amount of qualitative evidence that *de novo* gene birth does occur is increasing [Cai et al., 2008, Baalsrud et al., 2017], the quantitative evidence about the frequency of this phenomenon is lacking. Synteny-based methods suggest that sequence divergence is not the main source of orphan genes [Vakirlis et al., 2019]. Purifying selection is expected to screen occasionally translated open reading frames (ORFs) in a way that makes them more viable as raw material [Wilson and Masel, 2011], thus making *de novo* gene emergence less implausible. The physico-chemical properties and secondary structures of evolved and random sequences are very similar, and randomly created sequences can be tolerated *in*

vivo by *Escherichia coli* [Tretyachenko et al., 2017]. Neme et al. [2017] showed that at least two non-coding and one protein-coding gene, and many more whose coding nature has not been tested, resulted from 150 randomly generated sequences (mimicking *de novo* evolution) in lab conditions. While the beneficial nature of these genes is disputed [Weisman and Eddy, 2017, Knopp and Andersson, 2018], substantial tolerance clearly exists.

The only way to be confident that a particular putative *de novo* TRG is not merely a rapidly evolving gene duplicate is to find evidence of how it emerged. If we can identify homologous DNA region(s) in the species outside the clade from which a gene has emerged (i.e. the outgroup species) and if these DNA regions are non-coding, then we have the evidence that the gene is specific to this particular clade, as well as information about the nature of the origination process. When a putative TRG has simply diverged beyond detection of its protein-coding homologs, no homologous non-coding sequence will be found (although a syntenic homologous coding sequence may be found upon close scrutiny), and so a false positive *de novo* gene identification will be avoided. A false positive could, however, arise from a horizontal gene transfer followed by pseudogenization in one lineage. Fortunately, such cases can often be excluded when homology to the donor clade is detectable. Both lack of donor sequence and pseudogenization in a member of the focal clade are required to generate such a false positive, a scenario that in combination should be reasonably rare. One important scenario to consider is when, following a gene duplication, the ortholog in the outgroup is lost or diverges beyond detectable ho-

mology. It is therefore important to consider all likely homologous DNA regions in outgroup species, not only the single most likely region. One way to do this is to check whether the identified region in the outgroup species is homologous to any other regions in that genome. This is made relatively easy when the duplicated DNA region contained flanking, better-conserved genes, such that local synteny information can be exploited.

Even with synteny, detecting homologous non-coding sequences can be difficult. Non-coding regions of the genome are either under little evolutionary constraint, or under constraint very different from that of protein-coding regions, depending on their function or lack thereof. What constraint they have might apply to very general properties, rather than to specific nucleotides at specific codon positions, and hence might not be enough to prevent rapid degradation of detectable homology [Frigola et al., 2017]. Most constraints acting on non-coding regions are not sequence-specific, often they are related to general genome organisation and/or methylation. This means that it is necessary to confine analysis to closely related genomes in order to identify evolutionary origins of TRGs. In genomes that are separated by larger evolutionary distances the sequence similarity between a *de novo* gene and a homologous non-coding genomic region in another species degrades beyond detectability. Hence the oportunity to distinguish between genes that emerged from previously non-coding genomic regions and genes that emerged by divergence from previously coding genes is lost. In their pioneering work Jain et al. [2019] introduced a measure of "evolutionary traceability" of a protein family that quantifies the evolutionary distance beyond which homologous proteins can no

longer be identified. For a protein of interest they collected its known homologs, and used a multiple sequence alignment and an evolutionary tree of those sequences to estimate the protein-specific indel rate and the parameter p of the geometric distribution characterising indel lengths. They also estimated these parameters for each of the known Pfam domains in the protein. These parameters (both protein- and domain-specific) were used to simulate the evolution of the protein of interest. The "evolutionary traceability" of the protein was subsequently measured by BLASTp at time steps of 0.1 substitutions per site. The advantage of this approach is that it aims to model protein evolution in a plausible way, accounting for additional constraints on folding domains. The main, though admittedly unavoidable, limitation of this approach is that the stringency of selecting homologous sequences for protein-specific parameter estimation directly influences the estimate of "evolutionary traceability" obtained. No similar work exists for homologous non-coding DNA regions.

Some analyses restrict their search for putative TRGs to the set of already-annotated protein-coding genes. Gene annotations are primarily based on ORF length, transcription, and homology to known genes. Hence, a short TRG that has no previously known homologs is likely to be missed by an annotation algorithm, despite the fact that TRGs are expected *a priori* to be short. An alternative approach is to start with all ORFs present in the genome and exclude the ones that have no evidence for being functional. Previous studies used different types of evidence of functionality: Blevins et al. [2017] analysed deep RNA sequencing and ribosome profiling data,

Ruiz-Orera et al. [2018] combined that with proteomics data and single nucleotide polymorphism analysis, while Vakirlis et al. [2017] developed a logistic regression classifier trained on coding and non-coding sequences using such properties as codon frequency, hydrophobicity and aromaticity scores and structural predictions (secondary structures, transmembrane and disordered regions). However, TRGs are expected to have a narrow expression profile [Wu and Knudson, 2018] and they may have sequence properties distinct from well-studied protein families. The best indication of functionality is sequence conservation [Graur et al., 2013], which is by definition unavailable for single-species TRGs, even when they are functional. There is thus a trade-off between false positives (non-functional ORFs) and false negatives (true TRGs excluded from the analysis). There is thus a trade-off between false positives (non-functional ORFs) and false negatives (true TRGs excluded from the analysis). Beginning with annotated protein-coding genes tilts the balance toward false negatives, while beginning with all ORFs tilts it toward false positives. Regardless of how stringent or relaxed the requirements for evidence of functionality are, the resulting set of putative TRGs is unlikely to be both high confidence and exhaustive, limiting the potential for novel biological insights. To advance our knowledge about *de novo* TRGs, resource-intensive experimental investigations of the most promising candidates are required.

Candidates need to be chosen from studies that prioritize avoiding false positives over avoiding false negatives. For example, BSC4, which is found only in *Saccharomyces cerevisiae* has synthetic lethal knockouts [Cai et al., 2008]. This strong functional evidence made it a good candidate for structural biology experiments,

which showed that it has a characteristic three-dimensional fold [Bungard et al., 2017]. Absent such direct experimental data as synthetic lethal screens, the best indication of functionality is sequence conservation between several species [Graur et al., 2013], which is by definition unavailable for single-species TRGs, even when they are functional.

Several studies have taken an approach similar to ours to investigate the evolutionary origins of putative TRGs (i.e. focusing on the evolutionary evidence of how they emerged) in primates, insects and rosids [Toll-Riera et al., 2008, Zhou et al., 2008, Wissler et al., 2013, Sun et al., 2015, Donoghue et al., 2011]. One of the major limitations of this type of study is that TRG candidates are extensively ruled out based on thinly justified *a priori* assumptions about TRGs, in some cases discarding up to 61% of candidate genes [Vakirlis et al., 2017]. In particular, some studies excluded genes with more than one coding exon because "it is difficult to distinguish the absence of coding potential due to frame-shifts and stop codons from the alternative explanation of evolutionary change of intron-exon boundaries" [Guerzoni and McLysaght, 2016]. It is also believed that the evolution of both a long ORF and an intron splicing signal is highly improbable [Knowles and McLysaght, 2009]. Interestingly, other studies excluded single coding exon genes either to avoid promoter- or enhancer associated transcripts (PROMPTS and eRNAs) [Ruiz-Orera et al., 2015] or to avoid possible contamination of TEs incorrectly annotated as genes [Toll-Riera et al., 2008]. Similarly, many studies excluded genes whose length is below a certain threshold [Yang and Huang, 2011], genes

with compositions too far from an average established protein-coding gene, and genes that are evolving too fast [Vakirlis et al., 2017]. In the most extreme case, Casola [2018] excluded TRG candidates which are present in several copies in a genome due to a belief that young genes could not have had the time to duplicate.

Once they have identified TRGs, the second major limitation of these studies is testing a set of candidates for each of the hypothesised mechanisms of origination sequentially instead of looking holistically at the evidence available for each of the genes to establish their evolutionary origin. *De novo* protein-coding genes might be born within functional ncRNA, within introns or alternative frames of older protein-coding genes, or from intergenic sequences. Despite our desire to classify new genes into discrete categories, the evolutionary journey from an ancestral sequence to a new protein-coding gene might involve multiple steps, or vary along the gene's length. For example, TRGs might contain both previously non-coding sequences and fragments of well-established genes. McLysaght and Hurst [2016] proposed the classification of TRGs into several groups based on the proportion of the sequence that has previously been under natural selection for protein-coding properties. However, the distinction can blur, e.g. if previously protein-coding genes are pseudogenised or rearranged into non-coding sequence (see a review by Balakirev and Ayala [2003]), and are then resurrected as part of a TRG. While pre-existing transcription may obviously be an advantage, most of the genome is likely to be transcribed across relatively short evolutionary time in at least one cell type [Neme and Tautz, 2016]. Non-functional transcripts of intergenic ORFs have been

hypothesised to be a reservoir of genomic raw material that can increase organism's ability to adapt [Brosius, 2005]. On the other hand, intergenic ORFs have lower %GC content, compared to alternative reading frames of existing genes that tend to be GC-rich. Lower %GC content makes intergenic ORFs less suitable as a raw material, because the polypeptides produced are more ordered and hence more prone to aggregation. This is discussed in detail in the following papers [Basile et al., 2017, Ángyán et al., 2012, Wilson et al., 2017, Casola, 2018, Foy et al., 2019].

1.2 Computational studies of genome regulation

A genome is a blueprint for life; the process of genome regulation allows a cell to grow, function, and differentiate. Bulk RNA sequencing (RNA-seq) has been used extensively to learn about the different ways in which cells respond to various stimuli, and about the differences between tissues in multicellular organisms. While bulk RNA-seq allows us to identify marker genes that are upregulated in certain tissues or conditions, it does not capture the heterogeneity between individual cells. As a result, we might observe two genes that are upregulated in a sample without knowing that no one cell expresses both genes simultaneously. For this reason bulk RNA-seq data is of limited utility for inferring gene regulatory networks and studying cell differentiation pathways. The development of experimental techniques for capturing gene expression profiles at a single cell level started with the pioneering work by Tang et al. [2009]. The techniques for generating single cell RNA-seq (scRNA-seq) data have been constantly improving ever since. The availability of

scRNA-seq data created many opportunities for biological discoveries that were not possible with bulk transcriptomics. To create a vivid impression of scRNA-seq an analogy comparing it to a fruit salad is often used, as opposed to bulk RNA-seq that is compared to a smoothie.

In this section I will first describe how scRNA-seq data is generated and how it is used to answer different biological questions. Subsequently, I will catalogue various sources of noise that contribute to the noisiness of scRNA-seq data and highlight the types of information present in the data. With this I will highlight the need for new computational methods to analyse scRNA-seq data, and the opportunities for generative neural network applications. I will review the advantages and limitations of recent efforts to apply generative neural networks to scRNA-seq data to set the stage for my work in this area.

1.2.1 Single cell RNA sequencing

Single cell RNA-seq (scRNA-seq) captures the transcriptional state of a cell. This transcriptional state is created by an underlying gene regulatory network in which a limited number of regulators (transcription factors, cofactors, etc.) influence each other and their downstream target genes. Many biological pathways that are active in the cell at the same time, both related to the cell identity and to the current activity of the cell, result in a transcriptional state that is an entanglement of numerous biological signals. The ability to measure the transcriptional state at a single cell

resolution allows us to answer previously unanswerable biological questions and poses previously unmet challenges of noisy compositional data. In this section I will explain how scRNA-seq data is generated and what questions researchers aim to answer with this data. I will provide a non-exhaustive overview of methods developed for scRNA-seq data analysis aimed at addressing these questions with a focus on showcasing the diversity of ideas behind the methods. For convenience I will hereafter use "real gene expression level" to mean the number of transcripts produced by a cell and "gene expression level" to mean the estimate produced by scRNA-seq.

1.2.1.1 scRNA-seq experimental protocol

A typical scRNA-seq is shown in Figure 1.1. All scRNA-seq protocols start with dissociating and isolating individual cells. This is followed by a library construction step - intracellular mRNAs are captured, reversetranscribed to cDNA and labelled with cell specific barcodes. Many experimental protocols also label captured molecules with unique molecular identifiers (UMI) [Islam et al., 2013] - six to eight nucleotide long random oligonucleotide barcodes attached to individual cDNA molecules during sequencing library preparation. UMIs allow us to distinguish between sequenced copies of distinct molecules and copies arising through PCR amplification. This is followed by a PCR amplification. The libraries are then combined and sequenced to a required depth. There is a choice between two types of sequencing techniques - fulllength mRNA sequencing and 3' enrichment

methods. After sequencing, the reads are demultiplexed to identify the cells of origin and are mapped to the reference genome or transcriptome. This results in read data. In UMIbased protocols the reads are further demultiplexed to produce counts of captured mRNA molecules, thus resulting in count data free from PCR amplification noise. See Ziegenhain et al. [2017] and Svensson et al. [2017] for a comprehensive overview of different protocols.

1.2.1.2 Explosion of scRNA-seq data analysis methods

Analysis strategies for scRNA-seq data fall into two broad categories - cell centric approaches and gene centric approaches. See Figure 1.2 for an overview of the scRNA-seq data analysis strategies. Cell centric approaches are aimed at identifying either cell types or cell trajectories. Clustering algorithms are usually used for cell type identification. This approach is aimed at explaining the heterogeneity in the data by categorising cells into non-overlapping groups. In data that captures cell trajectories - either differentiation trajectories or dynamic response to a stimulus - cells cannot be divided into separate clusters and graph based methods for trajectory inference are used instead. Both lines of investigation depend on defining an association measure between the expression profiles - either a distance or a similarity measure. Since most measures don't work in 15000+ dimensions (a typical number of genes in a dataset) Euclidean distances in a small number of dimensions defined by principal component analysis (PCA) are often used. Gene centric approaches are aimed at using the heterogeneity between the cells in the

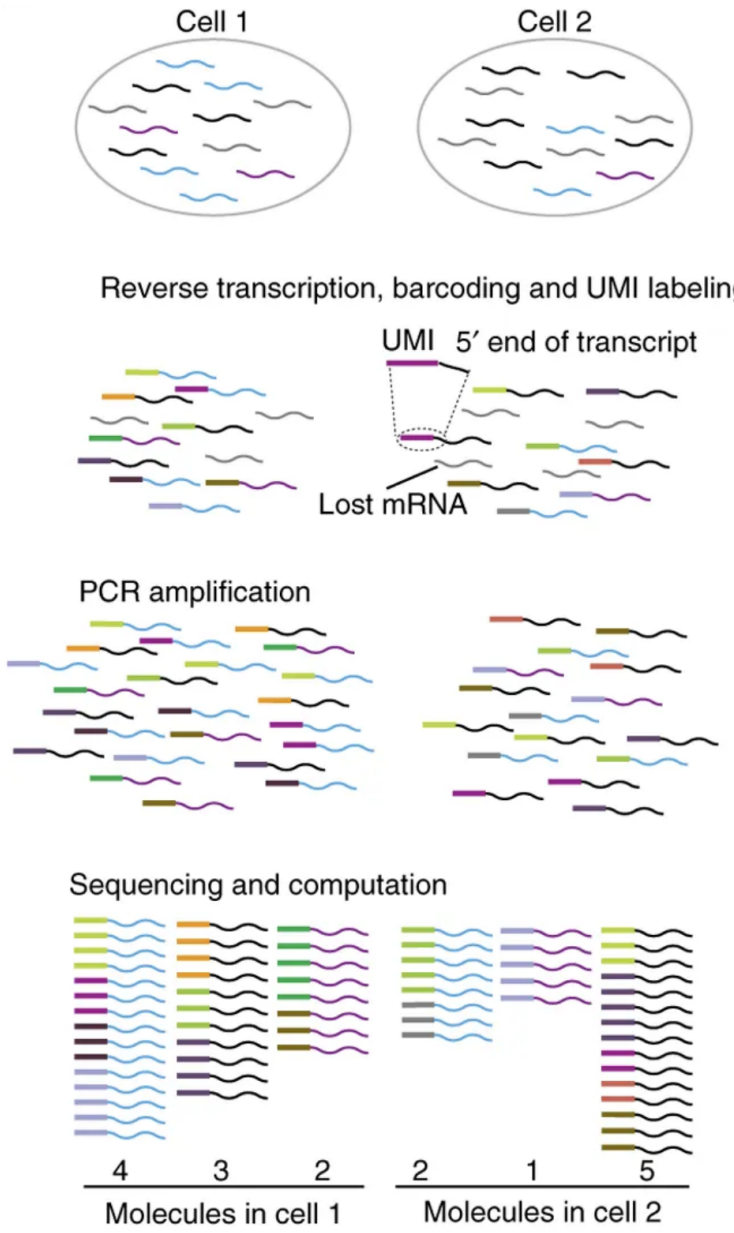


Figure 1.1: The steps in a typical scRNA-seq protocol. The figure is adapted from the work by Islam et al. [2013].

data as a context in which gene expression is to be understood. These approaches are focused on identifying the drivers of gene expression patterns - either through differential expression analysis or gene regulatory network inference.

Regardless of what analysis will be performed, data is usually subjected to quality

Cell centric methods	Cell type analysis - these methods assign cells into a finite number of non-overlapping clusters. They are based on clustering (k-means) or community detection algorithms (k-nearest neighbors).
	Cell trajectory analysis - these methods reconstruct the continuous process underlying the data. They are based on graphs or diffusion maps.
Gene centric methods	Differential gene expression analysis - these methods identify expression fold changes between cell sub-populations. They are based hurdle or Bayesian models.
	Gene regulatory network analysis - these methods infer regulatory relationships between genes. They are based on gene co-expression measures defined in terms of correlation, mutual information, or via a regression model.

Figure 1.2: scRNA-seq data analysis strategies.

control and pre-processed first. See Luecken and Theis [2019] for a comprehensive overview of typical scRNA-seq data quality control and pre-processing steps. After pre-processing, the data is used as an input into an algorithm for performing a specific analysis task. Numerous algorithms originally created for bulk RNA-seq data have been re-purposed to be applicable to scRNA-seq data, and many more algorithms have been designed specifically for scRNA-seq data. These methods have been selectively reviewed [Bacher and Kendziorski, 2016, Camara, 2018, Tanay and Regev, 2017] and exhaustively catalogued by many members of the single cell community. The two main catalogues of scRNA-seq data analysis tools list 177 (`https://github.com/seandavi/awesome-single-cell`) and 485 (`https://www.scrna-tools.org/`) tools. Despite the number of tools being well into hundreds, most studies analysing scRNA-seq data prefer to use

established R packages that have been used in many previously published studies. Seurat [Butler et al., 2018, Stuart et al., 2019] and Monocle [Trapnell et al., 2014, Qiu et al., 2017b] are very popular. These packages perform many different analysis tasks and often preclude the user from using pre-processed data as input thus retaining control over which pre-processing steps are performed.

1.2.1.3 Cell type analysis methods

Most cell type identification methods start with calculating a distance or a similarity measure between all pairs of gene expression profiles in the dataset. The clusters are subsequently identified by a clustering algorithm with or without an *a priori* defined number of clusters, or by a manual inspection of the data visualised in two dimensions using a dimensionality reduction algorithm. Popular clustering algorithms like k-means are rivalled by community detection algorithms like knearest neighbours which only consider neighbouring (i.e. similar) cell pairs as potentially belonging to the same cluster. Unlike randomly initiated clustering algorithms, community detection algorithms operate on a greatly reduced search space and are therefore faster and more readily scalable to large amounts of data. To address the challenges of defining a distance or a similarity measure in extremely high dimensionality, Xu and Su [2015] proposed an algorithm based on the concept of shared nearest neighbour which takes into account the surroundings of neighbouring data points. To ensure the robustness of a clustering algorithm applied to noisy data Kanter et al. [2018] proposed the “cluster robustness score”. It is calculated

by adding increasing amounts of noise with zero mean and increasing variance to the data and identifying clusters that can withstand the most noise. They showed that manually identified biologically meaningful cell clusters have high robustness scores, and clusters resulting from over-fitting are the ones most sensitive to noise.

Once the cells are assigned to clusters, cell types present in each cluster are identified via marker genes. Since the construction of clusters relies on a distance or a similarity measure that is heavily influenced by highly expressed genes, all resulting clusters by design have identifiable marker genes associated with them. Luecken and Theis [2019] showed that they can identify significant marker genes even when clustering random data generated by Splatter [Zappia et al., 2017]. PanoView [Hu et al., 2019a] is an iterative clustering method that is aimed at identifying both major and rare cell types simultaneously. It first identifies the most confident cell clusters with a density-based clustering algorithm applied to the first three principal components in the data, and then iteratively repeats the clustering with the remaining cells in a new principal component space. The unsupervised methods for data clustering ignore prior knowledge of marker genes and transcription factors (TFs) associated with cell types known to be present in the data. Instead, Zhang et al. [2019a] proposed a semi-supervised method that uses known marker genes to identify cell types with an expectationmaximisation algorithm. If marker genes are not known *a priori*, Aibar et al. [2017] proposed a method that first identifies TFs and genes they might regulate and then clusters cells based on binary matrix of "regulon" activity. Xie et al. [2019] proposed an alternative to clustering that does

not involve a distance or a similarity measure - a fully connected neural network, similar to the ones used for image classification, with the number of output nodes equal to the expected number of cell types.

1.2.1.4 Cell trajectory analysis methods

Isolating and studying individual cell types was previously possible with bulk RNA-seq, though at a much lower resolution. Capturing differentiation trajectories has only become possible with the advent of single cell technologies. Trajectory inference methods interpret scRNA-seq data as a snapshot of a dynamic process and the properties of this process are investigated. At this point, it is important to remember that not all biological pathways are regulated at the transcriptional level. A cell's "decision" to follow one of the downstream trajectories available to it might be externally induced or stochastic in nature. In both of these scenarios the gene expression profile does not contain the relevant information. A recent example is work by Baser et al. [2019] that shows that differentiation of adult neural stem cells in mice is post-transcriptionally controlled. They demonstrate that stem cells translate abundant transcripts with little discrimination, the onset of differentiation results in increasing regulation of the translation process, and hence differentiating cells become increasingly dependent on post-transcriptional control.

Trajectory inference methods reconstruct the continuous process underlying the data by identifying paths through the space spanned by the gene expression profiles.

These paths are constructed by minimising the transcriptional differences between neighbouring cells. Trajectory inference methods are based on the assumption that the gene expression profiles change smoothly in time both within cells and between the parent and the descendant cells. This assumption is reasonable when studying processes that take place on time scales comparable with mRNA synthesis and degradation, as the number of mRNA molecules varies smoothly at these time scales. Another strong assumption made by trajectory inference methods is that cells can never move backwards along the trajectory or switch path. The assumptions made by the methods limit the scope of questions that can be answered using them. For example, an important question about differentiation is whether the cells go through smooth, continuous progressions of transcriptional activity during the fate transitions or whether the fate transitions occur in a discontinuous, stochastic manner whereby signals modulate the probability of the transition events and cells must actively "jump" to overcome barriers between discrete fates [Moris et al., 2016]. This could mean that the fate transitions are characterised by a peak in gene expression variability [Richard et al., 2016]. The methods that assume smooth, continuous progressions of transcriptional activity during the fate transitions cannot answer these questions. Most trajectory inference methods focus on genes with smoothly varying expression levels. Identifying these genes is the only way to find out whether the identified trajectories are related to differentiation, cell cycle progression, circadian rhythm, or some other process. In cells, multiple biological processes are inevitably occurring simultaneously and hence to isolate the trajectories of interest it is desirable to regress out the biological effects of other processes.

Wanderlust [Bendall et al., 2014] was one of the first trajectory inference methods, only able to detect linear non-branching trajectories. Monocle [Trapnell et al., 2014] revolutionised the field by introducing the concept of "pseudotime" and a trajectory inference method that was able to produce trajectories that are bifurcating (a single trajectory that splits into two) and multifurcating (a single trajectory that splits into two or more trajectories). The concept of pseudotime is really simple - a single cell is selected as a "root cell" and the rest of the cells are ordered in terms of their distance from the root cell (based on a pre-defined distance or a similarity measure). The root cell corresponds to "time = 0". The groups of cells that are x distance away from the root cell correspond to "time = x", where x is continuous. Pseudotime was intended as a quantitative measure of progress through a biological process that allowed ordering of cells along the discovered trajectories. Unfortunately, it is often interpreted as a proxy for developmental time, which it is not. Monocle 2 [Qiu et al., 2017b] was subsequently built based on the hypothesis that an explicitly defined trajectory structure will result in more robust pseudotimes and branch assignments, but this method does not perform well in benchmarking studies.

Another major stepping stone in this field was the application of diffusion maps - work by Haghverdi et al. [2015] that was subsequently refined into Scanpy tool [Wolf et al., 2018]. The concept of diffusion maps was first introduced by Coifman et al. [2005] as a nonlinear data summarisation technique. Dimensions in a

diffusion map highlight the heterogeneity of different cell populations, thus emphasising transitions in the data. In the context of trajectory inference, each cell has its diffusion radius around its measured position, and this cell's ability to diffuse is described by an isotropic Gaussian wave function. This leads to a definition of a probability of one cell transitioning into another cell - the interference of the two wave functions that is itself a Gaussian wave function. Haghverdi et al. [2016] introduced the simple idea of plotting histograms of the pseudotime coordinates to identify dense regions which likely correspond to preferred transcriptomic states that are more stable than other transient states. Currently there are more than 70 trajectory inference methods available. A useful way to group these methods is based on whether a specific trajectory topology is assumed by the algorithm or can be suggested by the user. See Saelens et al. [2019] for a review of these methods and a proposed way to classify them into different types, Cannoodt et al. [2016] for a review of earlier methods and Saelens et al. [2019] for a recent comprehensive benchmark of 45 methods.

1.2.1.5 Gene centric analysis methods

Gene centric approaches to scRNA-seq data analysis are aimed at identifying either differentially expressed genes or gene regulatory networks. SCDE [Kharchenko et al., 2014] and MAST [Finak et al., 2015] were the first tools developed specifically for differential expression analysis in scRNA-seq. In contrast to similar methods designed for bulk RNA-seq data, these methods try to account for the

technical dropouts (genes that were expressed in the cell but not captured by the experiment) in scRNA-seq data. SCDE is a Bayesian method that estimates the likelihood of a gene being expressed in each sub-population of cells and the likelihood of expression fold change between those sub-populations. MAST is a hurdle model - a two-part generalised linear model that specifies one process for zero counts (technical dropouts with non-detectable expression) and another process for positive counts (strongly non-zero expressed genes). MAST adjusts for the fraction of genes expressed in a cell interpreting it as a proxy for both technical and biological sources of variation. edgeR [Robinson et al., 2009] and DESeq2 [Love et al., 2014] are popular methods designed for differential expression analysis of bulk RNA-seq data - they assume that the data follows a negative binomial count distribution. Independent reviews by Jaakkola et al. [2016] and Soneson and Robinson [2018] showed that they can be successfully applied to scRNA-seq data and are not outperformed by tools designed specifically for single cell data.

Gene regulatory network inference methods are based on gene coexpression measures defined in terms of correlation, mutual information, or via a regression model. Chen and Mar [2018] benchmarked three gene regulatory network inference methods developed specifically for scRNA-seq data (SCODE [Matsumoto et al., 2017], PIDC [Chan et al., 2017], SCENIC [Aibar et al., 2017]) alongside five methods developed for bulk RNA-seq data. They showed that the networks inferred by different methods vary substantially and in general methods developed specifically for scRNA-seq data infer networks that are less similar to each other. An indepen-

dent review by Pratapa et al. [2019] benchmarked twelve gene regulatory network inference methods and again showed that the networks inferred are considerably inconsistent with each other. They also observed that methods that don't require pseudotime inference often performed better. In my opinion there are two main reasons that can explain why most methods for gene centric scRNA-seq data analysis perform poorly. First, most of these methods assume that gene regulatory relationships are the same in all cells in the dataset, which might not be the case. Second, none of these methods account for the fact that scRNA-seq data is compositional data.

1.2.1.6 Discoveries powered by scRNA-seq data

Despite the challenges associated with scRNA-seq data analysis, this type of data has yielded an impressive list of discoveries. Projects based on scRNA-seq data have made important contributions in many different areas, from developmental and regeneration biology to studies of disease, ageing, and the immune system. In the field of developmental biology, Yu et al. [2019] studied pancreatic development, Moignard et al. [2015] identified transcriptional programs that underpin organogenesis, and Cao et al. [2019b] created the Mouse Organogenesis Cell Atlas that provides a global view of developmental processes in mice. Nowotschin et al. [2019] analysed all endoderm populations within the mouse embryo until midgestation and defined the transcriptional architecture that accompanies the emergence of the first endodermal population and pluripotent epiblast lineage. Cao et al. [2019a]

constructed cell-lineage maps and provisional gene networks for 41 neural subtypes that comprise the whole larval nervous system of *Ciona intestinalis*. The review by Griffiths et al. [2018] highlights other areas of developmental biology where scRNA-seq data have made important contributions. In the field of regeneration biology, Qin et al. [2019] classified cell states throughout the adult axolotl limb regeneration process and Siebert et al. [2019] identified differentiation trajectories for each cell lineage in Hydra - a cnidarian that is capable of whole-body regeneration from a small piece of tissue. In the field of immune system studies, Jordão et al. [2019] characterised the mouse brain's innate immune system and Masuda et al. [2019] studied the endogenous immune system of the central nervous system in mice during development and disease. Hodge et al. [2019] characterised the cell types involved in human cerebral cortex architecture and identified their equivalents in mice thus extending the potential impact of other studies conducted in mice. Paul et al. [2015] studied production of all of the cellular components of blood and blood plasma in mice and Guiu et al. [2019] showed that all cells of the mouse intestinal epithelium contribute actively to the adult intestinal stem cell pool. The review by Moignard and Göttgens [2016] highlights the contributions of scRNA-seq data to our understanding of stem cell differentiation. Aizarani et al. [2019] constructed the Human Liver Cell Atlas using samples from nine healthy human donors. Wang et al. [2019c] identified gene signatures associated with immune cell exhaustion during HIV infection and Velmeshev et al. [2019] found that pathways affected by autism regulate both synapse function and neural outgrowth and migration. Hu et al. [2019b] created a map of disease-related genes for human fetal retinal cells

and highlighted the importance of retinal progenitor cells as potential targets of inherited retinal diseases. de Soysa et al. [2019] identified mechanisms of disrupted cardiac development, thus providing a framework for investigating congenital heart defects. In the field of ageing, Dulken et al. [2019] characterised interactions between T cells and neural stem cells in old mice brains and Kimmel et al. [2019] showed that while both cell types and tissue environments influence the magnitude and the trajectories of ageing, the influence of the cell identity is predominant and hence different cell types age in unique ways.

1.2.2 Noise and information in scRNA-seq data

As Laurence Hurst put it, "for my own part, the questions of interest remain understanding whether genomic activity is mostly so much noise and rubbish or all part of some poorly understood but exquisite machine" [Cheifet, 2019]. In this section I will first catalogue the sources of noise that are intertwined with each other and with the signals in scRNA-seq data. I will review the types of methods often used to mitigate these sources of noise and their limitations. Keeping the noise structure of the data in mind, I will then discuss the signals present in scRNA-seq data. Finally I will explore the implications of the often overlooked fact that scRNA-seq data is compositional data.

1.2.2.1 Sources of noise in scRNA-seq data

Compared to bulk RNA-seq data, scRNA-seq data is incredibly noisy. Sources of noise that are present in the data can be broadly divided into two categories - biological and technical. Biological sources of noise include properties of transcription, cell-specific properties (cell size, cell cycle stage, mutational load, stress response, etc.) and gene-specific properties. Technical sources of noise depend on the experimental protocol used and the methods used for mapping the data to a reference genome or transcriptome. Another often overlooked source of noise is time. The mRNA content in the cell results not only from the current cell identity and the cell activity programs currently executed, but also from the presence of residual mRNAs from the previous cell state and/or activity, and the lag between a cell making a decision and actually generating enough mRNAs to execute this decision. See Figure 1.3 for a summary of the noise sources. There are two main reasons why dealing with noise in scRNA-seq data is challenging. First, these many different sources of noise are entangled and hence can not be identified individually. Second, there is no ground truth and therefore no straightforward way to benchmark methods aimed at mitigating the noise in the data.

The current expression level of a gene in a cell, i.e. the number of mRNA molecules, is governed by transcription bursts and mRNA degradation. Even if the cell is maintaining a constant level of expression of a gene, due to transcriptional bursting the number of mRNAs fluctuates over time [Bartman et al., 2016]. See Raser [2005] for an early review of studies of the stochastic and inherently random nature of

Biological	Technical	Time
• Properties of mRNA transcription (transcription bursts, competition for ribosomes) and degradation • Cell-specific properties (cell size, cell cycle stage, mutational load, stress response) • Gene-specific properties (methylation, transcript stability, rate of protein synthesis)	• Batch effects • Gene specific (based on length, %GC content, etc.) mRNA loss and degradation, efficiency of mRNA capture and reverse-transcription to cDNA • library size and quality • PCR amplification bias or sequencing errors in UMIs • sequencing protocol and sequencing depth • limitation of the sequencing step - only a small proportion of transcripts in a cell is sequenced • methods used for mapping the data to a reference genome/transcriptome (gene-ambiguous read handling)	Number of mRNAs in a cell is determined by • current state (cell identity, activity programs executed) • presence of residual mRNAs from the previous state • lag between a cell making a decision and actually generating enough mRNAs to execute this decision

Figure 1.3: Sources of noise in scRNA-seq data can be categorised in three groups.

gene transcription, and reviews by Raj and van Oudenaarden [2008] and Koster et al. [2015] for a detailed overview of how the dynamics underlying transcriptional regulation have been studied across different domains of life. Raj and van Oudenaarden [2008] highlighted that characteristics of the stochastic process of transcription depend both on the biophysical parameters governing the gene transcription and on the gene regulatory network structure. [Donovan et al., 2019] simultaneously tracked TF binding and transcription at one locus, revealing the timing and correlation between the binding of a particular TF and the subsequent transcription. Based on their results they proposed a model in which multiple RNA polymerases initiate transcription during one burst as long as the TF is bound to DNA, and bursts terminate upon TF dissociation. Li et al. [2018] experimentally validated that for TFs that regulate transcription burst frequency, as opposed to

amplitude or duration, weak binding is sufficient to elicit strong transcriptional responses. They also showed how refractoriness of a gene after a transcription burst enables rapid responses to stimuli, that might hinder the trajectory inference method performance as the smoothness assumption is violated.

Other sources of biological noise in scRNA-seq data are cell properties that differ amongst the cells in the dataset. These include cell size, cell cycle stage, mutational load and produced stress response. Foreman and Wollman [2019] showed that in fact the majority of expression variability results from differences between cells and that the remaining variability is effectively at the Poisson limit, meaning that the contribution of transcriptional bursting is relatively minimal. Buettner et al. [2015] showed that cell cycle related variation in gene expression is not restricted to known cell-cycle marker genes - 44% of moderately to highly variable genes that have not previously been associated with the cell cycle show a significant correlation with at least one known cell-cycle marker gene. Catala and Elela [2019] showed that in *Saccharomyces cerevisiae*, instead of mRNA transcription, it is the mRNA degradation that plays a lead role in promoting cell cycle-dependent gene expression by triggering promoter-dependent co-transcriptional RNA degradation. Gene expression plasticity is necessary for a cell's ability to accommodate for mutational load, and this plasticity adds to the noise. El-Brolosy et al. [2019] studied transcriptional adaptation in response to mutations in zebrafish and mice, and demonstrated that alleles displaying mutant mRNA decay also exhibit upregulation of genes that share sequence similarity with the mutated gene, suggesting

a sequence-dependent mechanism. In contrast, alleles that fail to transcribe the mutated gene do not exhibit transcriptional adaptation and give rise to more severe phenotypes. Another major contributor of cell-specific noise is associated with a cell's stress response induced by the sample acquisition, the dissociation protocol and subsequent manipulations. To avoid this, Saint et al. [2019] suggested snap-freezing cells immediately after harvesting, thus fixing both the cell morphology and the transcriptome. Beliakova-Bethell et al. [2013] and Richardson et al. [2015] examined the effects of fluorescence activated cell sorting (FACS) separation on cell transcriptomes and found no significant effect. The cell's size, current cell cycle stage and produced stress response are all linked to the cell's identity and hence cannot be disentangled, at least not in a straightforward way.

Several studies captured gene-specific properties that could be associated with gene-specific noise sources in scRNA-seq data. For example, Horvath et al. [2019] found that methylated genes in *Arabidopsis thaliana* have lower expression noise levels than unmethylated genes. An additional source of noise comes into play in analyses that use scRNA-seq data as a proxy for protein abundance levels, since protein synthesis is a stochastic process that is not independent from mRNA synthesis but is not directly correlated with it either. Apart from the number of transcripts, also their stability should be taken into account. Newman et al. [2006] monitored protein levels at a single-cell resolution in yeast and demonstrated that the noise in protein abundance is dominated by the stochastic processes of mRNA production and degradation. They also showed that there are protein-specific differences

in noise that are strongly correlated with a protein's mode of transcription and a protein's function. For example, proteins that respond to environmental changes exhibit noisy levels of abundance, and proteins involved in protein synthesis exhibit a much more stable level of abundance. Zarai and Tuller [2018] examined the effect of competition for ribosomes in *Saccharomyces cerevisiae* and demonstrated that periodically changing the mRNA levels of a single gene leads to the translation of all genes being affected in a periodic manner. This non-independence between genes and important roles that mRNA degradation plays, for example as shown by Catala and Elela [2019] and El-Brolosy et al. [2019], results in unexpected additional sources of noise.

The main source of technical noise results from the limitation of the sequencing protocols - only a small proportion of transcripts in a cell is sequenced in any given experiment. Due to mRNA loss and degradation, and inefficiency in the reverse transcription reaction, sometimes the proportion of mRNAs that are sequenced is as low as 10% Qiu et al. [2017a]. Other sources of technical noise include mRNA capture efficiency, library size and quality, efficiency of reversetranscription to cDNA, PCR amplification bias, specifics of the sequencing protocol used and sequencing depth. For UMI-based protocols noise from PCR amplification is not a concern, though UMIs themselves have been recognised as a source of noise as their 6-8 nucleotides long sequences are prone to sequencing errors. These errors lead to an enrichment in the fraction of UMIs separated by small Hamming distances and all associated with a single gene. To prevent these sequencing errors from influencing

the count data, Smith et al. [2017] proposed a network-based method that leverages the differences between the distributions of average edit distances amongst random sets of UMIs and UMIs associated with a single gene. Properties of genes (length, %GC content, etc.) also contribute to noise as they lead to gene-specific affinity to mRNA capture and reversetranscription to cDNA. Data from fulllength mRNA sequencing has to be normalised for gene length, while data produced with 3' enrichment methods is gene length independent. While in general rates of technical dropouts are higher for lower expressed genes [Hicks et al., 2017], Kharchenko et al. [2014] showed that the rates of technical dropouts as a function of expression magnitude is different for different cells. Noise associated with batch effects occurs when cells are handled in distinct groups. The slightly different environments experienced by the cells in different batches result in differences in cell transcriptomes, i.e. a biological source of noise. These differences also result in variations in measurements of the transcriptome (efficiency of mRNAs capture and reversetranscription to cDNA, library characteristics, PCR amplification bias and sequencing depth), i.e. a technical source of noise. The main limitation inhibiting many potential studies based on acquiring scRNA-seq data is that more often than not the batch effects coincide with biological differences being studied, for example different time points in development. Mapping of the reads to a reference genome or transcriptome results in noise associated with reads mapping to several genes. Most mapping tools simply discard gene-ambiguous reads, while methods like RSEM [Li and Dewey, 2011] and Alevin [Srivastava et al., 2019] account for both gene-unique and gene-ambiguous reads.

Grün et al. [2014] analysed the sources of technical noise in scRNA-seq data and showed that sampling noise is a dominant source for genes expressed at low levels, while variability in sequencing efficiency due to batch effects is a dominant source for genes expressed at high levels. Kim et al. [2015] used RNA spike-ins to quantify how the observed variation in expression of a gene is split between technical and biological noise. They concluded that a large fraction of the variation in the expression of a gene can be explained by technical noise sources, especially for genes expressed at low levels, and only about 20% of observed variation is attributable to biological noise.

1.2.2.2 Noise mitigation in scRNA-seq data

Numerous methods have been developed to mitigate individual sources of noise or their combinations. As previously mentioned, since there is no ground truth there is also no straightforward way to benchmark these methods. Often synthetic data is the only way to assess the performance of a method. However, this assessment is likely to be biased as synthetic data is not compositional, and it is generated based on untestable assumptions about real data. Another problem is that the methods developed for mitigating noise in scRNA-seq data often themselves make untestable assumptions about the data.

Mitigating the noise in the data first requires an assumption about the underly-

ing distribution of the data. Which discrete distribution is the most appropriate for scRNA-seq data is still an open question. An ideal distribution would not only fit the data well but would also be interpretable, computationally tractable and compatible with biologically plausible assumptions. The Poisson distribution is the simplest distribution for handling count data, but it is not suitable for scRNA-seq data as it assumes that the variance is equal to the mean. Non-zero values in scRNA-seq data are overdispersed, i.e. their variance grows faster than the mean, and hence the negative binomial is a better fit. Amrhein et al. [2019] examined the fit between the mechanistic transcription-degradation models and commonly used discrete probability distributions. Their results indicate that the negative binomial distribution arises as a steady-state distribution from a canonical two-state promoter-activation model of transcription bursting. Grün et al. [2014] investigated the distribution of transcript counts and found that the negative binomial distribution explained the distribution for the largest fraction of genes. scRNA-seq data is heteroscedastic, i.e. there are sub-populations that have different variances, and, more generally, the distribution shape of the data is very different in different parts of its large dynamic range. This suggests that a mixture of distributions would be most appropriate for this data. For example, a continuous mixture of Poisson distributions where the mixing distribution of the Poisson rate is a gamma distribution results in the negative binomial. Lopez et al. [2018] showed that the addition of a zero-inflation component is important for explaining a subset of the zero values in scRNA-seq data, and that it captures important aspects of technical variability that are not represented by the negative binomial distribution. Grønbech et al. [2018] proposed an

analysis based on Bayesian model selection. Their results show that both negative binomial and zero-inflated negative binomial are a good fit for the data. Poisson-beta distributions have been suggested as an alternative [Delmans and Hemberg, 2016, Vu et al., 2016]. Vu et al. [2016] showed that Poisson-beta distributions are a good fit for scRNA-seq data and they can capture the transcription burst frequency and size as well as the expression drop-off caused by technical noise. Amrhein et al. [2019] advocate the use of the negative binomial distribution as it provides a good trade-off between computational complexity and biological simplicity.

In this section I will cover the major categories of methods that have been proposed for mitigating the effects of noise sources in scRNA-seq data. The most common way to mitigate the noise in scRNA-seq data is to normalise the data. Often the normalised values are calculated by dividing the values in the data by cell-specific scaling factors based on the library size, often referred to as "cell size factors". This approach is based on the assumption that the total amount of mRNA is constant across all cells in the data. McGee et al. [2019] showed that the performance of normalisation methods is reduced when there are large changes in the total amount of mRNA per cell. Another commonly used assumption is that the expression of most genes does not vary across cells. For example, popular methods like DESeq [Anders and Huber, 2010] and TMM [Robinson and Oshlack, 2010] assume that about 50% of the genes are not differentially expressed between cells. An alternative assumption used by scPLS [Chen and Zhou, 2017] is that genes in the data can be classified into two groups - a control set that is independent from the predictor

variables and target genes that are of interest in this specific analysis. McGee et al. [2019] showed that both of these assumptions result in misleading conclusions if there are large shifts in expression profiles across the data. Linnorm [Yip et al., 2017] is a recently introduced method that identifies genes with constant expression throughout the data and calculates the normalisation parameters based on them. Lun et al. [2016] proposed a different approach where expression values are first summed across pools of cells, and then these summed values are used to compute pool-based size factors, which are deconvolved to produce cell-based factors. The deconvolution is based on normalising the cell pools against an average reference and repeatedly splitting the cells into pools to create a linear system of equations. Summing the gene expression values in every pool alleviates the presence of problematic zeros in the data. Again, this method is based on the assumption that less than 50% of the genes can be upregulated in the data and less than 50% of them can be downregulated. Azizi et al. [2017] argued that this type of normalisation removes important cell type-specific information and proposed BISCUIT - a Bayesian probabilistic model that learns cell-specific parameters by iteratively normalising and clustering the cells, thus aiming to separate noise from biological signals. No matter how sophisticated an algorithm, simply scaling the data is not sufficient for normalisation as it does not address the systematic non-linear biases in the data. To both normalise the data and stabilise the variance, thus removing the influence of technical characteristics while preserving biological heterogeneity, Hafemeister and Satija [2019] proposed a model based on the Pearson residuals from regularized negative binomial regression, where the library size is used as a covariate in a gener-

alized linear model. To address the tendency of an unconstrained negative binomial model to overfit the data, they pooled information across genes with similar expression levels to obtain stable parameter estimates. Bacher et al. [2017] recognised that scRNA-seq data shows systematic variation in the relationship between gene-specific expression and sequencing depth, and a single cell-specific scaling factor cannot accommodate for this. They showed that common normalisation methods overcorrect weakly and moderately expressed genes and under-normalise highly expressed genes. Instead, they proposed SCnorm that uses quantile regression to estimate the dependence of a gene expression on the sequencing depth for every gene. A second quantile regression is used to estimate scaling factors for each of the groups of genes with similar dependence values. Within-group adjustment is then performed using the estimated scale factors to regress out differences in the sequencing depth while accounting for potential differences across groups of genes. Adding spike-in transcripts to the lysate prior to library construction enables a different class of normalisation methods that utilise the observed expression values of the spike-in transcripts with known abundance. The spike-in transcripts are added as naked RNA, and thus they may be degraded and reverse transcribed to cDNA at different rates from endogenous mRNAs. Because of that, the usage of spike-ins has become less prevalent even though this technique was initially very popular. McGee et al. [2019] pointed out that this criticism of spike-ins does not take into account the compositional nature of the data. They showed that the performance of downstream methods improves if spike-ins are used in a compositional manner - even if the spike-ins have different properties from the endogenous mRNAs they

help to minimise misleading conclusions from downstream analyses. Athanasiadou et al. [2019] proposed a method using a spike-in count table together with a vector of true spike-in abundance to estimate the library calibration factor, which is then used to estimate nominal abundances of endogenous mRNAs in a cell. See Vallejos et al. [2017] and Cole et al. [2019] for a comprehensive review of normalisation techniques. Luecken and Theis [2019] cautioned that no perfect data correction method exists and inevitably the values in the data will be over- and/or undercorrected, thus both reducing the background variation in the data and altering the variances of the gene expression profiles in an unintended way. This hinders the performance of the downstream analysis methods, as genes whose background variation is over-corrected are more likely to be identified as differentially expressed. If experimental design is taken into account by the normalisation method, i.e. signals that do not conform to it are treated as noise, then normalisation leads to an overestimate of the effect size. Luecken and Theis [2019] concluded that at least for gene-centric analyses of scRNA-seq data normalisation should be avoided.

While data normalisation is aimed at mitigating technical noise, there have been also numerous efforts to mitigate biological sources of noise, most notably the effects of the cell cycle phase. Buettner et al. [2015] showed that merely removing the set of annotated cell-cycle marker genes from the data is not an option, since cell cycle related variation in gene expression is not restricted to marker genes. Their results suggest that 44% of moderately to highly variable genes that have not previously been associated with the cell cycle show a significant correlation with

at least one known cell-cycle marker gene. Instead, the expression values in each individual cell should be normalised appropriately based on the identified cell cycle phase of each cell. The cell cycle phase of a cell is usually identified based on the expression of known marker genes, for example the ones listed by Macosko et al. [2015]. Often this normalisation is performed either via a simple linear regression against the cell cycle score or with a more complex mixture model. See Scialdone et al. [2015] for a benchmark of methods for cell cycle phase prediction. McDavid et al. [2016] argued that the cell size variation accounts for the transcriptomic effects generally attributed to the cell cycle phase, and hence normalising for these effects based on the cell cycle score leads to the unintended biases in downstream analysis. In general, at any given point in time many biological processes occur within the same cell, and these processes are not independent from each other. Therefore, correcting for one process may unintentionally mask the signal of another. Luecken and Theis [2019] suggested that regressing out several sources of technical and/or biological noise should be performed in a single normalisation step that accounts for dependences between these covariates. They cautioned against performing several normalisation steps in a sequential manner.

As discussed earlier, more often than not the batch effects in the data coincide with biological differences being studied. This means that a batch effect correction method should be able to distinguish between variation in the data that is attributable to batch effects - which is an entanglement of both biological and technical sources of noise - and variation that is attributable to properties of interest. Similarly to data

normalisation, simply scaling the data cannot mitigate the non-linear batch effects. Any batch effect correction method that uses all cells in a batch to fit the batch parameters will confound the batch effect with biological differences being studied. The two most popular methods for batch effect correction are Seurat [Butler et al., 2018] and MNN [Haghverdi et al., 2018]. Seurat uses canonical correlation analysis to learn a shared gene correlation structure that is conserved between the batches. It then identifies rare populations of cells that may be non-overlapping between the batches, i.e. the cells that cannot be well described by this shared gene correlation structure. The method based on mutual nearest neighbours (MNN) proposed by Haghverdi et al. [2018] does not rely on equal population compositions across batches; it is based on a much weaker assumption that there is at least one cell sub-population that is shared between the batches. See Büttner et al. [2018] for a comprehensive benchmark of batch effect correction approaches. They proposed a k-nearest-neighbour batch-effect test for quantification of batch effects and used it to assess the methods. Recently several interesting new methods have been proposed. Stanley et al. [2018] introduced an approach based on diffusion maps that aligns the batches onto a shared data manifold. Gong et al. [2019] proposed a method based on a simple assumption that the cells from different batches that are mutually close to each other are more likely to belong to similar cell types. It is an improvement on the method proposed by Haghverdi et al. [2018] where the distances between cells from different batches are compared to the background distribution of cell distances (the null model) instead of being considered in isolation. Wang et al. [2019d] proposed BERMUDA - an autoencoder based method.

Prior to using the data to train the autoencoder, it clusters the cells in each batch individually and identifies similar clusters across different batches. The autoencoder's loss function consists of two parts - the reconstruction loss and the transfer loss. Similar clusters from different batches are merged by minimising the transfer loss, which is calculated by estimating the differences between the distributions of low-dimensional embeddings corresponding to the pairs of similar clusters across different batches. BERMUDA does not correct the data, instead it provides a latent space embedding of the data that can be used for downstream analysis as it is free from batch effects.

A notable property of scRNA-seq data is the abundance of technical dropouts present in the data - the genes that were expressed in a cell but not captured by the experiment. The amount of technical dropouts is minimal in bulk RNA-seq data and hence new methods had to be developed specifically for scRNA-seq data. These methods have two challenges - distinguishing between technical dropouts and the real lack of expression, and imputing appropriate expression values for the technical dropouts. An appropriate expression value is often defined as the mean of the distribution from which this value would have originated, conditional on the gene being expressed. The dropout imputation methods can be broadly divided into two groups - data smoothing based methods and model based methods. Data smoothing based methods aggregate information from similar cells - MAGIC [van Dijk et al., 2018] using data diffusion, and knn-smooth [Wagner et al., 2017] using k-nearest neighbour smoothing algorithm. Model based methods make an

assumption about the distribution of the data - SAVER [Huang et al., 2018] imputes all zero values in the data, while scImpute [Li and Li, 2018] aims to distinguish between technical dropouts and genes that are not expressed, and only impute the values of technical dropouts. Andrews and Hemberg [2019] benchmarked several data smoothing and model based methods, as well as an autoencoder based method DCA [Eraslan et al., 2019]. To evaluate the risk of generating irreproducible differentially expressed genes and false positive correlations between genes, they used simulated datasets and real datasets with their values permuted. They showed that data smoothing based methods generated many false positives, while the performance of model based methods varied depending on the diversity of cell types in the dataset. Andrews and Hemberg [2019] highlighted SAVER as the method that is least likely to generate false or irreproducible results. I will summarise five methods that were not included in this benchmark, mostly due to their recent publication date, to demonstrate the breadth of diversity in approaches that have been suggested for addressing the technical dropout problem. RESCUE [Tracy et al., 2019] generates many subsets of highly variable genes by sampling with replacement, and subsequently clusters the cells based on the corresponding gene expression signatures in each of the subsets. In contrast to statistical methods, Chen and Varshney [2019] proposed a geometric method based on optimal recovery - an approximation-theoretic approach for estimating linear functions of a signal. netSmooth [Ronen and Akalin, 2018] is a method based on network diffusion; it uses priors for the covariance structure of the gene expression profiles in order to smooth the expression values. DeepImpute [Arisdakessian et al., 2018] is a method

based on a deep neural network; it uses dropout layers to learn the patterns in the data. DECODE [Mohammadi et al., 2018] uses co-occurrence of genes in the cells to infer a gene dependency network. The local network neighbourhood of each gene is subsequently used to distinguish between technical dropouts and genes that are not expressed. To impute technical dropouts, DECODE builds a predictive model based on activity patterns of each gene's most informative neighbours. Despite the growing arsenal of methods for dropout imputation, fundamentally all of them can be divided into two groups. The methods in the first group aim to impute only zero values in the data, and to do that they use the information in the non-zero values. This results in corrected data that has unintended statistical properties. Prior to dropout imputation, small non-zero expression values in a cell are likely to correspond to genes that are expressed at a higher level than the genes with expression values equal to zero. After the imputation, the zero values of the technical dropouts have increased, while the values corresponding to lowly expressed genes that were detected have not. An intuitive property of the data - that genes with higher expression values are likely to have been more highly expressed in a cell - is now distorted. The methods in the second group avoid this by not limiting the data correction to zero values only. In these methods, information is pooled from all values in the data, and based on that information many or most values in the data (not only zero values) are corrected. This is problematic in a different way. To circumvent the limitations inherent to both groups of methods, Leote et al. [2019] proposed a method for improving dropout imputation quality by integrating the information about relationships between genes learned from available data other

than the scRNA-seq dataset being analysed.

The currently available methods for data normalisation, cell cycle and batch effect mitigation, and technical dropout imputation are imperfect. Let us make an unrealistic assumption that perfect methods do exist. Using them would result in a scRNA-seq dataset that is free from cell- and gene-specific effects, as well as cell cycle and batch effect and contains no technical droputs. This dataset would still be very noisy because of the stochastic processes that are ongoing in a cell (mRNA synthesis and degradation, etc.) and because of the stochastic nature of all steps in the protocol (mRNA capture, mRNA reversetranscription to cDNA, etc.). To address this, data denoising methods have been proposed. These methods aim at capturing cell population structure and gene interaction networks from the data, and using them to infer cell- and gene-specific parameters for the denoising process. Several methods that fit a probabilistic model for each gene measurement in each cell are available [Risso et al., 2018, Pierson and Yau, 2015, Prabhakaran et al., 2016]. They are based on the assumption that a generalised linear model can be used to accurately map onto a lower dimensional manifold underlying the data. These methods correctly treat the data as overdispersed count data and account for technical dropouts. The usual limitations apply - the performance of these methods cannot be benchmarked in a straightforward way due to lack of ground truth, and they make untestable assumptions about the data, for example that the relationship between a data point and its representation in a lower dimensional space is linear. Additionally, scaling these methods to large datasets is often computationally

infeasible. Recently Peng et al. [2019] proposed SCRABBLE - a method based on the mathematical framework of matrix regularisation. This method optimises a complicated objective function that consists of three terms. The first ensures that imputed values for genes with non-zero expression remain as close to their original values as possible. The second ensures that the rank of the imputed expression matrix is as small as possible, thus incorporating the assumption that only a limited number of distinct cell types is present in any given dataset. The third uses a user supplied bulk RNA-seq dataset that originates from the same species and the same tissue. This third term ensures the consistency between the average gene expression of the aggregated imputed data and the average gene expression of the bulk RNA-seq dataset. In contrast to these numerous efforts to denoise the data, several other methods assumed a different strategy. They simply binarise a scRNA-seq dataset, thus retaining only the noise-free information about which genes have been detected in a cell. Li and Quon [2019] proposed scBFA - a method for cell type identification and trajectory inference from binarised data. This method is motivated by their observation that dimensionality reduction based only on the gene detection measurements performs better than if both detection and quantification measurements are used. It assumes that due to the bad signal to noise ratio, gene expression quantification is uninformative; gene detection is relatively less noisy. Qiu [2018] proposed a cell clustering method based on evaluating the co-occurrence between pairs of genes using the chi-square statistics. Xie et al. [2019] proposed a method based on a neural network trained on binarised data.

1.2.2.3 Information in scRNA-seq data

The scRNA-seq data contains gene expression profiles of individual cells. Each expression profile is an entangled mixture of expression patterns associated with cell identity and numerous activity programs ongoing in a cell, for example cell division, differentiation, responses to environmental cues, etc. The expression of each gene is governed by a limited number of regulators (TFs, cofactors, etc.) through a gene regulatory network. This is an evolved network, which means that its properties might be different from the expectations associated with a designed network. Expression values in the data correspond to the number of experimentally captured transcripts from a single cell, but, due to the limitations of currently available protocols, the proportion of transcripts that are sequenced is sometimes as low as 10% Qiu et al. [2017a]. Consequently, the expression values cannot be analysed directly as absolute measures. There are three (overlapping) types of information contained in a single expression profile. First, an expression profile contains a non-exhaustive list of expressed genes, but many genes that are not highly expressed are missing from this list. Second, it contains a (more complete) set of highly expressed genes, for which it happens rarely that not a single transcript is captured. Third, it contains relative information based on the assumption that more highly expressed genes are likely to have more sequenced transcripts associated with them. Analysis methods that are able to pool the information from numerous expression profiles contained in a dataset are essential for interpreting the scRNA-seq data.

The fundamental property of the scRNA-seq data is that it is compositional. This

property is rarely acknowledged, and the majority of the methods applied to the scRNA-seq data do not account for it. A compositional data point can be represented by a positive real vector. The space spanned by these vectors is called a simplex. The information contained in a data point is a set of ratios between the components. Therefore, a compositional data point is invariant to multiplication by a positive constant. This property implies that the sum of the values in a data point is uninformative. Aitchison [1982] published the landmark paper relating to compositional data. He argued that the statistical analysis of compositional data is inhibited by the lack of applicable concepts of independence, and the lack of rich enough parametric classes of distributions in the simplex. An R package for compositional data analysis [van den Boogaart and Tolosana-Delgado, 2008] and an unpublished scikit-bio Python package (`https://github.com/biocore/scikit-bio`) provide computational frameworks for working with compositional data that were previously not available.

Fernandes et al. [2014] was one of the first to advocate for treating all biological data composed of counts of a large number of features as compositional data, and not as count data. Similarly, Quinn et al. [2017] produced work aimed at fostering the adoption of methods suitable for compositional data in biological data analysis. Most recently McGee et al. [2019] highlighted that scRNA-seq data is compositional, and that this implies that the sum of values in an expression profile is uninformative. The sum is an artefact of the sampling procedure, and is unrelated to the absolute number of transcripts in a cell. The scale of the distances

between the values in an expression profile is not absolute but relative, and hence is only meaningful proportionally. This suggests that the scRNA-seq data analysis methods that interpret values in the data as absolute counts of mRNAs, even if they account for a complex noise structure in the data, are unlikely to produce valid results. Fernandes et al. [2014] cautioned that properties of compositional data are very different from datasets composed of numbers that can take any value, and hence standard statistical methods that assume the independence of the underlying observations are not applicable.

An important property of compositional data is that values in a data point, an expression profile in this case, are not independent. The value of one feature restricts the values of other features. An increase in the observed expression of one gene, i.e. more of its transcripts being sequenced, leads to less of other gene transcripts being sequenced even if the expression levels of all genes stay constant. This has two implications. First, the expression values are not directly comparable between genes in a cell or between cells. Second, observed correlations between genes might be simply a consequence of this non-independence property. To illustrate this, Aitchison [1982] gives an example where changing the abundance of one feature in a composition results in correlation between the others changing from strongly positive to strongly negative. This non-independence between the values in an expression profile has crucial implications when some of the genes are removed from the analysis. Aitchison [1982] showed that taking a sub-composition of a compositional data point often results in a completely different interpretation of the

correlation structure. Fernandes et al. [2014] noted that this effect is problematic because popular 16S tag-sequencing analysis tools require that reads falling below a certain threshold to be filtered out, and because ribosomal transcripts in RNA-seq are often removed either chemically or computationally. Most scientists, this author included, don't have an intuition for compositional data. It is tempting to interpret an observed value in an expression profile simply as a random variable drawn from the binomial distribution, where the number of trials (n) is the total number of gene's transcripts present in a cell, and the success probability for each trial (p) is the capture probability conditional on cell-, gene- and protocol-specific properties. It is important to remember that this is a wrong way to think about the scRNA-seq data.

Identifying relationships between variables, genes in this case, in compositional data is tricky. Lovell et al. [2015] cautioned that it is essential that only the ratios of the expression values are regarded as informative, and that measures of association like Pearson correlation, rank correlation and mutual information are inappropriate. Figure 1 of their paper [Lovell et al., 2015] illustrates that the correlations between relative abundances contain absolutely no information about the relationship between the absolute abundances that gave rise to them, due to the many-to-one mapping. They show how three gene pairs with correlations between their absolute abundances equal to +1, -1 and 0 all can look the same when only their relative abundances are measured. Based on the logratio variance $\mathrm{Var}(\log(x/y))$ originally proposed by Aitchison [1982] as a measure of association for variables that carry

only relative information, Lovell et al. [2015] proposed a related "goodness-of-fit to proportionality" statistic. This approach focuses on "strength" of proportionality and thus allows for comparison of relationships between different pairs of mRNAs without testing the hypothesis of proportionality directly. Lovell et al. [2015] showed that it can be used to meaningfully and interpretably assess the extent to which a pair of random variables are proportional. Erb and Notredame [2016] showed in a mathematically rigorous way that, when dealing with compositional data, choosing a single reference gene (whose expression is assumed to remain unchanged) as a scaling factor introduces spurious proportionality amongst the other genes. I believe that the fact that gene-centric scRNA-seq data analysis methods ignore the properties of compositional data is the reason why these methods perform poorly when benchmarked. Until mathematical and statistical theory about compositional data is developed (which would be a monumental scientific effort), I do not see the opportunities for gene-centric scRNA-seq data analysis.

1.2.2.4 Association measures

One of the main challenges associated with scRNA-seq data that gets almost no attention in the literature is finding a measure of association between two expression profiles. All scRNA-seq data analysis methods that don't rely on machine learning techniques instead rely on an association measure that can be calculated for any pair of gene expression profiles. An association measure can be either a distance or a similarity, or a dissimilarity (often defined as 1 - similarity). The Euclidean distance

is a very popular choice of association measure, despite the fact that it is guaranteed to fail in a high dimensional setting of gene expression profiles that often include 20000 genes or more. Beyer et al. [1999] explored the effect of dimensionality on the "nearest neighbour" problem. They showed that as dimensionality increases, the distance to the nearest data point approaches the distance to the furthest data point, and that this effect can occur already after 10-15 dimensions. Ronan et al. [2016] showed in an intuitive way why high-dimensionality of the data makes all points roughly equidistant from one another when using common distance metrics - three standard deviations cover 99.7% and 99.2% of the data in 1 and 3 dimensions respectively, the value drops to 97.3% in 100 dimensions and to 6.7% in 1000 dimensions.

The field of science concerned with scRNA-seq data analysis is very new, but the problem of defining an association measure between two highly dimensional profiles with many zeros is not new. For many decades ecologists have been looking for the best ways to compare habitats, where each one is characterised by a profile of numbers of spotted individuals from a long list of all species considered in a study. The properties of this type of data are very similar to scRNA-seq data - it is compositional data and the number of observed individuals (i.e. captured mRNAs) is expected to be much lower than the actual number. Most notable contributions include the work by Chao et al. [2004] that proposed a way to adapt Jaccard and Sorensen indices to account for species abundance, and the work by Clarke et al. [2006] that developed a zero-adjusted BrayCurtis coefficient. Despite the hope that

they might, these measure do not perform very well when applied to scRNA-seq data. I found a single example of Bray-Curtis similarity measure being used in a scRNA-seq data analysis method [Kharchenko et al., 2014].

Instead of trying to define an association measure in 20000+ dimensional space, it is desirable to reduce the number of dimensions. The simplest way to do so is to reduced the number of genes that are considered. Choosing a small set of highly variable genes is a popular approach, but it has been shown that methods for highly variable gene discovery are often not reproducible and produce inconsistent results [Yip et al., 2018]. An alternative approach is based on the manifold assumption - the assumption that high dimensional scRNA-seq data has a much lower intrinsic dimensionality. This assumption is driven by the fact that the space of biologically possible cellular states is much smaller than the space of all possible combinations of gene expression levels. This lower-dimensional manifold is thought to arise from the constraints of the underlying gene regulatory network. To find this lower-dimensional manifold, a dimensionality reduction method is required. For example, Burkhardt et al. [2019] used the Euclidean distance between 100 principal components (PCs) as a distance metric. However, it is well understood that principal component analysis (PCA) seeks to find the direction of the largest variance, and hence each component is a mixture of biologically unrelated processes [Tan et al., 2016]. These components can not be interpreted in a biologically meaningful way. Alternative methods for dimensionality reduction from thousands to a manageable number have been developed since DNA microarrays, many of those have recently

been applied to scRNAseq. Different latent variable models, both unsupervised (such as principle component analysis [Misra, 2002], independent component analysis [Engreitz et al., 2010], and non-negative matrix factorisation [Kim, 2003]) and semi-supervised (network component analysis [Liao et al., 2003], semi-supervised non-negative matrix factorisation [Gaujoux and Seoighe, 2012]) have been applied to analyse transcriptomic data, with an aim to represent the states of latent pathways using latent variables.

Kim et al. [2018] compared different similarity metrics and their influence on scRNA-seq data clustering. They evaluated distance measures (Euclidean, Manhattan and maximum distances) and similarity measures (Pearson and Spearmans correlation) by measuring the correspondence between the known cell clusters and the cell clusters computed using these measures. Their results demonstrated the importance of a similarity metric. They concluded that correlation-based similarity measures performed better on scRNA-seq data than distance measures. Subsequently, Skinnider et al. [2019] evaluated 17 measures in a similar manner. They showed that Euclidean distance (a measure widely used in many scRNA-seq data analysis methods) is one of the worst performing measures, while the proportionality-based measures performed the best, in line with the reasoning of Quinn et al. [2017]. In attempt to avoid the problems with an association measure, Faust et al. [2012] employed an ensemble approach. They computed Pearson and Spearmans correlation, as well as Bray-Curtis and Kullback-Leibler measures of dissimilarity, and defined an association measure based on all four of the com-

puted metrics. Needless to say that a combination of different metrics that do not relate to the underlying structure of the data is expected to be meaningless. A promising new direction is based on the work by Coifman and Lafon [2006] that shows that many dimensionality reduction methods are special cases of a general framework based on diffusion processes. The work by Wolf et al. [2018] is the first successful example of applying diffusion distances in a trajectory inference method.

In my opinion, the main criterion for a suitable association measure should be as follows. Given two identical copies of the same cell (which of course is impossible), the association measure should show that the distance between the two corresponding expression profiles is zero, or at least much smaller than a distance to any other cell. I was unable to construct such a measure. In general, it is unreasonable to assume that a single value of an association measure will be useful for any subsequent analysis. In terms of biological interpretation, an association measure between two cells can be defined as the relatedness of the two cell types, the overlap between the set of processes currently ongoing in the two cells, the similarity of the conditions the two cells are exposed two, etc. In my work I will propose a novel approach to defining an association measure. The main idea of this approach is to first reduce the dimensionality of the data using machine learning and subsequently select a set of generated dimensions appropriate for a particular type of downstream analysis.

1.3 Intended contribution of my work

The intended contribution of my work on *de novo* gene evolution is identifying the best candidate genes for experimental investigation, since this field is at the limits of what can be achieved computationally. I aim to strike the balance between evolutionary distance and proximity, that enables the inference of functionality and the identification of the evolutionary origins respectively. By identifying the evolutionary origins of candidate genes I intend to both validate their *de novo* origin and improve our understanding of how *de novo* genes emerge.

The intended contribution of my work on applications of generative neural networks (GNNs) to single cell RNA sequencing (scRNA-seq) data is to propose a new method that could address the limitations of existing methods discussed in this chapter. Instead of comparing autoencoders with a simple yet remarkably useful Principal Component Analysis (PCA), that is used extensively to reduce the dimensionality of scRNA-seq data, I start from the mathematical framework of PCA and modify it in simple steps to show the link between the familiar method and the newly adopted GNNs. I aim to provide a useful assessment of the information flow through an autoencoder and a comprehensive analysis of a latent space embedding created by an autoencoder. Using the knowledge gained, I intend to explore the constituents of the process of training an autoencoder on the scRNA-seq data - the internal architecture of the autoencoder, the dynamics of the autoencoder training, the data itself and the inherent noise in the data and randomness of the neural networks.

I intend to provide methods that are able to analyse scRNA-seq data, without making assumptions about the data that are fundamentally flawed. The prioritised properties of the proposed method are scalability and reproducibility. The amount of scRNA-seq data being generated is growing, and so are the opportunities to integrate previously generated data. Therefore methods that are able to perform the analysis on streaming data and take advantage of GPU-accelerated computations are essential. My work is aimed at enabling better outcomes of future studies based on scRNA-seq data and at contributing to the adoption of clinical applications of this type of data, which is currently inhibited by the lack of reproducible and assessable analysis methods. Pioneering work by Kort et al. [2019] included scRNA-seq analysis of a diverse group of 38 critically ill patients experiencing circulatory collapse as a common endpoint to wide ranging diseases and demonstrated the clinical applicability of this data. The contribution of my work is designed to grow in the future - as single cell technologies are developing rapidly and using machine learning is becoming a dominant strategy for analysing highly dimensional data and integrating different modalities of the data to foster biological discovery.

Chapter 2

Evolutionary origins of TRGs in *Drosophila* subgenus

This chapter investigates the evolutionary origins of *de novo* taxonomically restricted genes (TRGs) in *Drosophila* subgenus. I chose this clade because it is experimentally tractable, good quality genome assemblies of five closely related species are available and these genomes are compact, $\sim$140Mb. My approach was based on conservative but strongly justified criteria to identify putative *de novo* genes among annotated protein-coding genes that have homologs in at least two of the three species in the *simulans-sechellia-melanogaster* clade. I aimed to avoid genome sequencing and assembly artefacts by focussing on *de novo* taxonomically restricted gene families (TRGFs) instead of singleton TRGs. I looked for TRGFs that emerged after the split of the *simulans-sechellia-melanogaster* clade from the *yakuba-erecta* clade and before the speciation of *D. simulans* and *D. sechellia*. I used open reading frame (ORF) conservation across several species as a proxy for functionality under the selected-effect definition [Graur et al., 2013], as the half-life

of a non-functional ORF is small given the probability of acquiring a stop codon by chance. A dN/dS signal of selection would be still stronger evidence for functionality, but short sequences in three closely related species do not contain enough information to reliably distinguish deviations from dN/dS = 1. First, because only 2 to 3 sequences are available, i.e. a very small sample size that is not statistically significant. Second, because these 2 to 3 sequences are also very closely related and have very little diversity. The contributions made in this chapter are as follows:

- A list of high confidence protein-coding genes that emerged *de novo* in *Drosophila* subgenus
- A detailed discussion about the types of information contained in closely related genomes and how this information can be best used to guide future experimental studies

2.1 Materials and Methods

2.1.1 Data

The genome assemblies for *D. melanogaster*, *D. simulans*, *D. sechellia*, *D. yakuba* and *D. erecta* were downloaded from RefSeq [Haft et al., 2017] along with the genome annotations [O'Leary et al., 2015]. The completeness of the protein sets was assessed using BUSCO [Waterhouse et al., 2017], using 2799 Hidden Markov Models (HMMs) of single-copy orthologs found in >90% of species in the order Diptera. Table 2.1 summarises genome statistics for each species.

Table 2.1: Genome assemblies of *Drosophila* subgenus species used in this study.

Species (RefSeq assembly accession)	Assembly size (Mbp)	Molecule count	N50 (Mbp)	Proteins	BUSCO (%)
D. simulans (GCF_000754195.2)	124.96	7	0.45	14179	98.6%
D. sechellia (GCF_000005215.3)	166.59	1	0.042	16467	92.2%
D. melanogaster (GCF_000001215.4)	116.52	8	19.48	13916	99.3%
D. yakuba (GCF_000005975.2)	165.71	8	0.12	14824	98.4%
D. erecta (GCF_000005135.1)	152.71	0	0.45	13605	99%

2.1.2 Homology predictions

I used OMA v2.2.0 implementation of the OMA algorithm [Altenhoff et al., 2017] with default parameters to infer groups of homologous genes across six genomes: five *Drosophila* subgenus species and *D. ananassae* as an outgroup. All the genes annotated as protein-coding in the assemblies described above were used, regardless of their length. Genes as short as 11 amino acids were included. I selected orthologous families with genes in at least 2 of the species in the *simulans-sechellia-melanogaster* clade and no genes outside this clade as putative TRGFs for further analysis. To investigate the evolutionary origins of these putative *de novo* TRGFs, I used *D. yakuba*, *D. erecta*, *D. ananassae*, *D. suzukii*, *D. pseudoobscura* and *D. miranda* as outgroup species. Only in *D. yakuba* and *D. erecta* the sequence conservation was sufficient to make any conclusions. All species whose proteins are part of RefSeq database were used to validate that putative *de novo* TRGFs don't have homologs outside of clade of interest.

2.1.3 Validation of putative TRGFs

Putative TRGFs were first validated with sequence similarity searches in amino acid space against all non-redundant proteins in the RefSeq database, using BLASTp v2.7.1+ [Camacho et al., 2009] with default parameters (word size equal to 3, BLO-

SUM62 amino acid matrix, no sequence masking). All hits with e-value $\leq 1e-03$ and covering $\geq 50\%$ of the query were considered. If every gene in a putative TRGF had at least one hit to the species outside the clade, the family was removed from further validation. To identify the presence of known protein domains, I used putative *de novo* TRG sequences to search against the Pfam v31 database [Finn et al., 2015] of HMMs of known protein domains. I used HMMER v3.1b2 [Eddy, 2011] to perform these sequence searches against HMMs. If a gene in a putative TRGF had a hit to any of the domains in the Pfam database, the family was removed from further validation.

Remaining putative TRGFs were validated with sequence similarity searches in nucleotide space against the five genomes in *Drosophila* subgenus, using BLASTn v2.7.1+ [Camacho et al., 2009] with default parameters (word size equal to 7, maximum number of target sequences equal to 500, searching on both strands). I did not use tools like FASTA3 [Pearson, 2000] that take into account synonymous codons or amino acid similarity because the homologous DNA sequences are protein-coding in some species but not the others. BLASTn makes no additional assumptions about the evolutionary constraints specific to the query sequence, and hence is most suitable tool for this problem. For each gene I used both the whole gene sequence and the set of coding sequences (CDSs) as a query. This approach ensures that hits to even very short CDSs are retained, while also using the information in the non-coding parts of the gene when the information contained in a short CDS is insufficient. All hits with e-value $\leq 1e-03$ and covering $\geq 50\%$

of the query (a whole gene or a CDS) were considered, and overlapping hits were amalgamated. In cases where the total number of hits exceeded 1000, I ordered the hits by e-value and selected the five best hits per species. Hits (including self-hits to the genes) were aligned with MAFFT v7.407 [Katoh and Standley, 2013] using E-INS-i algorithm that makes minimum assumptions about the nature of the resulting alignment. I used the "–adjustdirectionaccurately" option to align hits located on different strands and the "–addfragments" option to subsequently add CDSs to the alignment of hits. Alignments were examined manually to remove the hits that were only covering parts of introns or untranslated regions (UTRs) and to extend promising hits that ended in the middle of the gene. After these amendments the remaining/extended hits were realigned and the resulting alignments were examined for presence of homologous ORFs in the *yakuba-erecta* clade. If an ORF was identified in at least one of the two outgroup species it was considered as evidence that the putative TRGF originated prior to the speciation of the *Drosophila* subgenus and the family was removed from further validation. Putative TRGFs that passed sequence similarity validations were manually examined for quality and consistency of annotations. I did not try to identify highly diverged homologs that are beyond detectability with BLASTp using more advanced methods like PSI-BLAST [Schaffer, 2001], HHMER [Eddy, 2011] or HHblits [Remmert et al., 2011] that rely on building a sequence profile. There were two reasons for that. First, given that the protein is only present in two species the resulting sequence profile would not contain much more information than a single sequence and hence it would be unlikely to yield useful results. Second, I relied on our assumption that

if a homologous gene is present in an outgroup genome it would be included in the BLASTn hits against that genome. At short evolutionary distances, DNA sequence conservation is likely to be no worse that amino acid sequence conservation due to limited average number of mutations per site. This assumption doesn't necessarily hold at large evolutionary distances, but for closely related species it would be extremely unlikely to identify a good DNA sequence match covering all of the gene and at the same time to miss a homologous gene that diverged beyond detectable similarity in nucleotide sequence space.

2.1.4 Inferring the origin of TRGFs

To infer the origins of TRGFs, I extracted genome annotations corresponding to the identified homologous DNA regions in all five species. I also identified homologous DNA regions in four additional species - two in the *melanogaster group* (*D. ananassae* and *D. suzukii*) and two in its sister clade *obscura group* (*D. pseudoobscura* and *D. miranda*). I used the BUSCA web server to predict protein subcellular localisation [Savojardo et al., 2018], TANGO to predict protein aggregation [Fernandez-Escamilla et al., 2004], and Wasabi for visualising multiple sequence alignments [Veidenberg et al., 2015]. All analysis was performed in Python v3.7.0, using packages biopython v1.73 [Cock et al., 2009] and gffutils v0.9.

2.2 Results

The five species of interest in the *Drosophila* subgenus (*D. melanogaster*, *D. simulans*, *D. sechellia*, *D. yakuba*, and *D. erecta*) had a common ancestor $\sim$8 Mya

[Obbard et al., 2012] (Figure 2.1). I look for taxonomically restricted gene families (TRGFs) that emerged after the split of the *simulans-sechellia-melanogaster* clade from the *yakuba-erecta* clade and before the speciation of *D. simulans* and *D. sechellia*. The divergence time between this clade and its sister clade that contains *D. ananassae* is ∼23 Mya. The intergenic sequence conservation between these clades is insufficient to distinguish between *de novo* TRGs (for which a homologous intergenic sequence in an outgroup genome is reqired) and highly diverged copies of well established genes. Apart from the five species clade shown in Figure 2.1 there are no other groups of five or more *Drosophila* species that are separated by less than 15 Mya. *D. melanogaster*, *D. simulans*, *D. sechellia*, *D. yakuba*, and *D. erecta* have genomes of ∼140Mb containing ∼14,000 protein-coding genes. There is no evidence of major segmental genome duplication in this clade, reducing complications in identifying homologous non-coding sequences. The genome assembly for *D. sechellia* is highly fragmented, as confirmed by N50 metric and a BUSCO estimate that ∼8% of the genes likely present in the genome are missing from the assembly (Table 2.1). The quality of the *D. sechellia* genome assembly leads to a different distribution of annotated protein lengths compared to other species in this clade (Figure 2.2). For this reason, gene loss can not be inferred based on the absence from *D. sechellia*.

Based on OMA homology inference algorithm [Altenhoff et al., 2017], these five *Drosophila* subgenus species contain 14,149 gene families. Amongst the inferred gene families there were 205 families with genes in at least 2 of the species in the

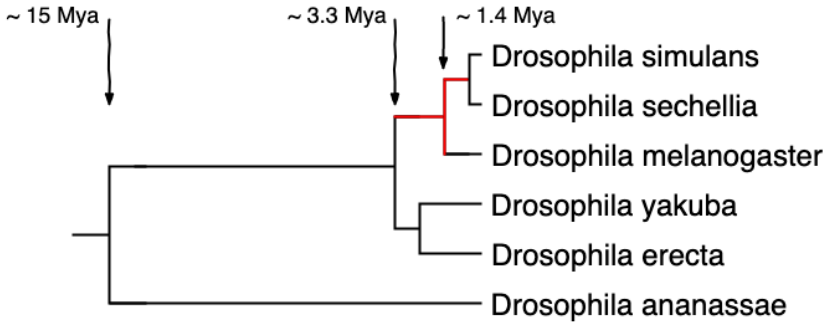


Figure 2.1: Species tree of the *Drosophila* subgenus. Branch lengths correspond to divergence time estimates by Obbard et al. [2012]. We looked for TRGFs that emerged during the evolutionary time marked in red, i.e. between ∼0.5 and ∼3.3 Mya. We ultimately confirm one TRGF shared only by *D. simulans* and *D. sechellia*, i.e. that originated between ∼0.5 and ∼1.4 Mya.

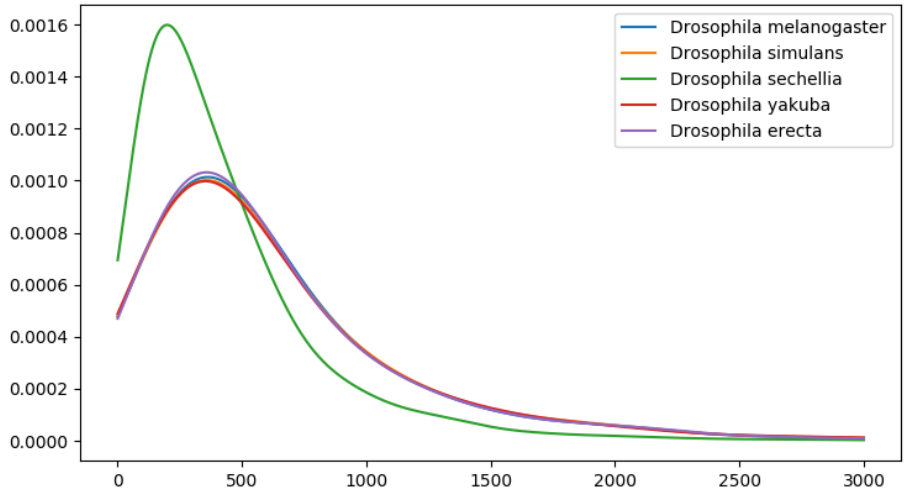


Figure 2.2: Protein lengths distribution in five *Drosophila* subgenus species. Number of amino acids is plotted on the x-axis.

simulans-sechellia-melanogaster clade and no genes from species outside the clade. Protein sequence similarity searches against the RefSeq database revealed diverged homologs outside the clade for 170 of these families. I used sequence similarity searches in nucleotide space to identify homologous DNA regions corresponding to the 35 putative TRGFs in all five genomes. Out of these 35 families, 18 contained conserved but unannotated ORF(s) covering $\geq$ 50% of the putative TRGF ORF

in at least one of the *yakuba-erecta* clade species, indicative of an earlier origin of these TRGs A conserved ORF in an outgroup was considered strong evidence that the gene family originated before the speciation of the clade. I was unable to obtain a continuous alignment of inferred homologous DNA regions in the *yakuba-erecta* clade for five putative TRGFs. It is unknown whether this is due to genome rearrangements and the lack of sequence conservation or simply because the true homologous DNA regions are missing from the genome assemblies. Only the 12 putative TRGFs for which I was able to obtain a continuous alignment of homologous DNA regions in all five species and show that the ORFs were only present in the *simulans-sechellia-melanogaster* clade were considered in further analyses.

Manual examination of genome annotations revealed problems and inconsistencies with ten putative TRGF annotations. For example, some of the genes were missing a start codon, contained overlapping CDSs or exons misaligned with splicing signals. In seven families the annotations were inconsistent across species - conserved homologous DNA regions formed a good alignment but gene annotations indicated different start/stop codons or splicing signals. These putative TRGF were removed from further analysis as they did not satisfy our requirement for a conserved ORF in more than one independently annotated species. Figure 2.3 summarises the different kinds of false positives that we eliminated.

To infer the evolutionary origins of the two putative TRGFs that remained following these filters, I looked at the homologous non-coding sequences whose common

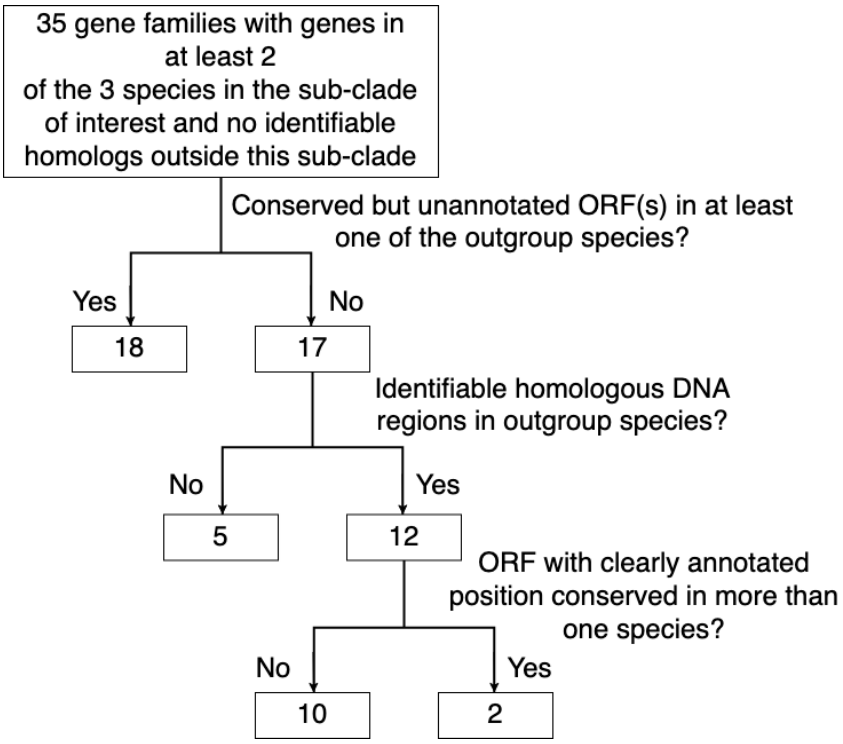


Figure 2.3: The elimination of TRGFs with either evidence of being false positive, or with insufficient evidence available.

ancestry with the TRGF preceded the origin of the TRGF. In the process, I was able to confirm the recent *de novo* status of the first, and refute that of the second.

The first TRGF evolved *de novo* in the *simulans-sechellia* clade on 3R chromosome, giving rise to *Dsim_GD19764* and *Dsec_GM10790*. These are annotated uncharacterised protein-coding genes with two CDSs and a conserved canonical GU—AG splicing signal. The protein is 129 amino acids long in *D. simulans* and 113 in *D. sechellia*. The conserved intron is 52 nucleotides long (not a multiple of 3), hence it is likely to pre-date the ORF (otherwise, later intronisation would have resulted in a frame-shift; see Yang and Huang [2011] for a detailed explanation). BUSCA predicts that this TRGF contains transmembrane alpha helix and

hence localises to the endomembrane system. There is transcriptomic evidence that *Dsim_GD19764* is expressed in the male reproductive system [Drosophila 12 Genomes Consortium, 2007], which is in line with previous results showing that TRGs are predominantly expressed in testes [Levine et al., 2006]. *Dsim_GD19764* and *Dsec_GM10790* have no functional annotations, neither experimentally established nor computationally infered. The N and C termini of *Dsim_GD19764* as well two segments of *Dsec_GM10790* are predicted to be disordered. Similarly, TANGO predicts that *Dsim_GD19764* and *Dsec_GM10790* have no regions prone to aggregation. This evidence is in line with the hypothesis that young *de novo* genes are often disordered since this trait is associated with not being prone to agregation, see Section 1.1.2 for details.

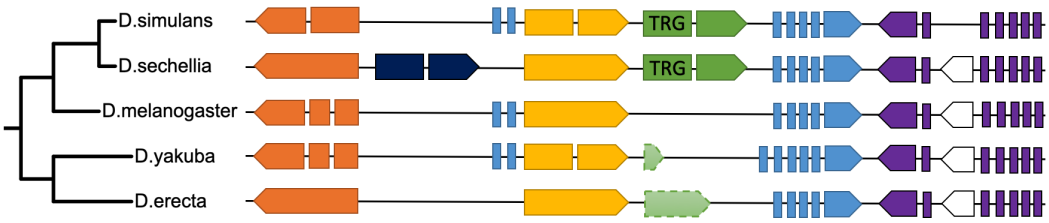


Figure 2.4: DNA regions homologous to the TRGF containing *Dsim_GD19764* and *Dsec_GM10790*. Homologous protein-coding genes are the same color (each element corresponds to an exon), small nuclear RNA (snRNA) genes are white. The direction of the arrow shows which strand the gene is located on. Features with dashed outlines are not annotated. The diagram is not to scale. In the order from top to bottom, the orange genes are *Dsim_GD29138*, *Dsec_GM10660*, *Dmel_CG12589*, *Dyak_GE25310*, *Dere_GG11200* (with a syntenic homolog *Dana_GF16073* in *D. ananassae*); the yellow genes are *Dsim_GD19763*, *Dsec_GM10789*, *Dmel_CG12590*, *Dyak_GE25451*, *Dere_GG12627* (with syntenic homologs *Dana_GF18925* and *Dpse_GA11706* in *D. ananassae* and *D. pseudoobscura* respectively); the blue genes are *Dsim_GD19765*, *Dsec_GM10791*, *Dmel_CG12591*, *Dyak_GE25452*, *Dere_GG12638* (with syntenic homologs *Dana_GF18926* and *Dpse_GA11707* in *D. ananassae* and *D. pseudoobscura* respectively); the purple genes are *Dsim_GD19639*, *Dsec_GM10658*, *Dmel_CG12161*, *Dyak_GE25306*, *Dere_GG11178*.

Dsim_GD19764 is located in an intron of a conserved protein-coding gene *Dsim_GD19765*, downstream of conserved protein-coding gene *Dsim_GD19763* located inside the same intronic region (Figure 2.4). In *D. sechellia* the *Dsec_GM10791* gene harbouring two genes inside its intron appears to have lost the first two exons and thus *Dsec_GM10790* is located in a similar genomic context but not inside an intron. The DNA regions that I presume to be homologous to TRGs in *D. melanogaster*, *D. yakuba* and *D. erecta* are located between the genes homologous to the ones neighbouring TRGs in the *simulans-sechellia* clade. The orange genes in Figure 2.4 have no functional annotations, but there is transcriptomic evidence that they are expressed in testes, antenna and mouth of adult organisms. The yellow genes are expressed in testes and head of adult organisms. The blue genes are expressed in wing disc during larval stage and in testes, brain and antenna of adult organisms [Cannavò et al., 2016]. This genes have two Ig-like protein folding domains. *Dmel_CG12591* in *D. melanogaster* is annotated with functions related to synapse organization and sensory perception of chemical stimulus. The purple genes are expressed in testes and head of adult organisms. These genes are annotated with functions related to proteasomal protein catabolic process and endopeptidase activity. Given that the TRG and all four genes surrounding it have transcriptomic evidence of being expressed in testes, it is plausible to conclude that this area of the genome is extensively transcribed in male reproductive organs in adult organisms.

There is too little nucleotide conservation for a good alignment to this region in

D. melanogaster, which contains no ORF. Alignment can be achieved with the *yakuba-erecta* clade, where the ORF is disrupted by an early stop codon. Since the ORFs in both *D. yakuba* and *D. erecta* are much shorter that the coding sequence of the TRG (the ORF in *D. yakuba* is only 5 amino acids long), these ORF are not considered to be potentially functional. Note that Hild et al. [2003] previously inferred a protein-coding gene located in this region on the opposite strand, but this gene is no longer part of the official genome annotations. I identified potential homologous DNA regions in four additional outgroup species (*D. ananassae*, *D. suzukii*, *D. pseudoobscura* and *D. miranda*), but the sequence conservation level was insufficient to establish the most likely ancestral state. These regions cover 50-65% of the TRG and have $\sim$70% sequence identity. Extending these hits in both directions did not lead to a better alignment. No start codon was present in these homologous DNA regions. We can thus rule out the possibility that two independent pseudogenization events, one in *D. melanogaster* and one in the basal lineage of the *D. yakuba-erecta* clade, created the illusion of a TRGF as a false positive. The homologous regions in *D. ananassae* and *D. pseudoobscura* contain three (orange, yellow and blue in Figure 2.4) and two (yellow and blue in Figure 2.4) syntenic homologs respectively, while the homologous regions in *D. suzukii* and *D. miranda* contain none.

The second gene family, that I mistakenly identified as a *de novo* TRGF, contains uncharacterised protein-coding genes *Dsim_GD20667* and *Dsec_GM19408*, and an unannotated homologous ORF in *D. melanogaster*. These annotated genes are

located on the 3R chromosome and contain a single CDS of length 155 in *D. simulans* and *D. melanogaster*. In *D. sechellia*, a frameshift close to the end of the CDS results in a conserved stop codon becoming in-frame and thus shortening the CDS to 139 amino acids. BUSCA predicts that the proteins localise in the nucleus.

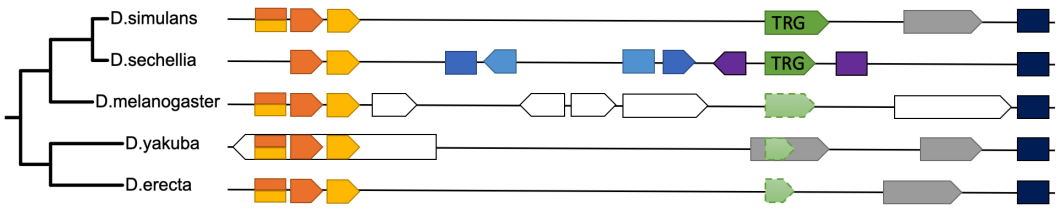


Figure 2.5: DNA regions homologous to the gene family containing *Dsim_GD20667* and *Dsec_GM19408*. Protein-coding genes are shown in colour, pseudogenes in grey and ncRNA genes in white. Homologous protein-coding genes are marked by the same colour, each element corresponds to an exon. Only the first of the seven exons of the dark blue gene is shown. The direction of the arrow shows which strand the gene is located on. Features with dashed outlines are not annotated. Genes shown directly above/below each other share sequence similarity. The diagram is not to scale.

These putative TRGs are located amongst protein-coding gene families syntenicly conserved in all five subgenus species, ∼70Kb downstream from a conserved pair of overlapping genes and ∼25Kb upstream from a conserved seven exon gene. The region between these two gene families is shown in Figure 2.5. A number of protein-coding genes are annotated in *D. sechellia* but have no detectable homologs in other species in the subgenus. *D. melanogaster* has a number of annotated ncRNAs, one of which overlaps with parts of *D. sechellia*-specific genes. Since these protein-coding genes are present in only one species I did not include them in our analysis as in the absence of conservation, I lack sufficient evidence that they are functional. The region containing the putative TRGs is annotated as an intron of

one of these *D. sechellia* protein-coding genes. The downstream region annotated as a pseudogene in *D. simulans*, *D. yakuba* and *D. erecta*, and as a ncRNA in *D. melanogaster*, is well-conserved in all species. The annotation boundaries vary among species.

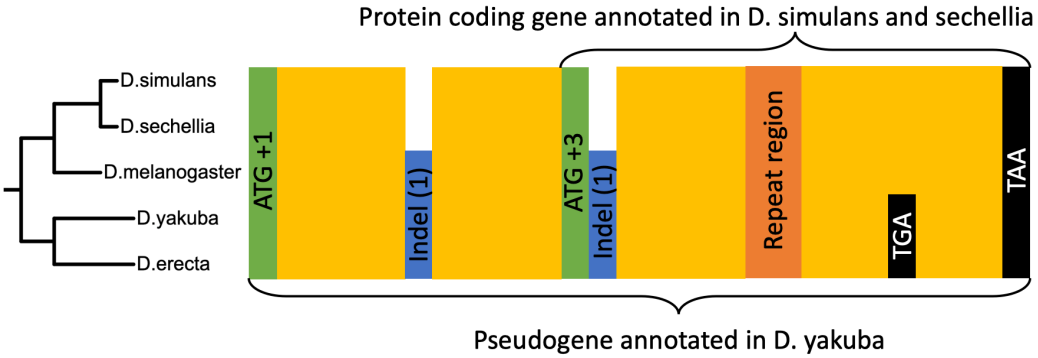


Figure 2.6: Sequence features of the ancestral ORF, which is annotated as a pseudogene in *D. yakuba*. Start codons of the annotated pseudogene and of the shorter putative TRG in the *simulans-sechellia-melanogaster* clade are in green, well conserved regions in yellow, frame-shift causing indels in blue, repetitive DNA in orange, and stop codons in black. I use the following frame numbering convention: the start codon is denoted the +1 frame, the other two frames on the same strand are denoted +2 and +3 frames. Frames of the start codons are marked relative to the pseudogene. The numbers in parentheses indicate how many more nucleotides (modulo 3) the species it is marked in has. The frames of the stop codons are not marked due to uncertainty about frame created by the repeat region. The two stop codons shown are located in the same frame.

The region containing the TRGF is extremely well-conserved in all five species and is annotated as a pseudogene in *D. yakuba*. Using BLASTn for similarity searches to identify the parent gene of this pseudogene I was only able to find a self-hit (i.e. hits to the genome region containing the putative TRG) and numerous matches covering $<10\%$ of the sequence in all species with an exception of *D. simulans* where I identified a 219 nucleotide long unannotated contig with 97.7% sequence identity. The similarity between the translated amino acid sequences is very weak. I was

unable to find any other evidence about the parent gene of this putative pseudogene.

The start codon of the putative TRGF is in a different frame than that of the *D. yakuba* pseudogene, suggesting that it evolved *de novo* in an alternative frame to the parent gene, but upon closer scrutiny I realised that this is not the case. Here by pseudogene I mean a genomic region that is annotated as a pseudogene. It is a sequence that evolved from a sequence of a protein coding gene, but no longer has an intact ORF. It is plausible that a TRG might evolve from a pseudogene, but the definition of *de novo* TRGs adopted here is that they evolved from previously non-coding regions of the genome. Since a pseudogene itself evolved from an ancestral protein coding gene, a *de novo* TRG can not evolve from a pseudogene. The start codon of the putative TRGF is flanked by two indels, which brings the frame of the annotated *D. yakuba* pseudogene in frame with the putative TRGF following its annotated start codon, see Figure 2.6. A *TG*-dinucleotide repeat region in the middle of the putative TRGF ORF appears to be poorly conserved; this could be either because of a genuinely higher mutation/indel rate, or merely because of a poor quality of reads/assembly in this region. The uncertainty created by this region and the fact that the length of the pseudogene is not a multiple of three makes it difficult to infer whether the putative TRGF shares the frame with the pseudogene throughout the whole sequence. More telling information comes from six stop codons conserved across the five species and located between the repeat region and the stop codon shared by both the pseudogene and the putative TRGF. Four stop codons are in +2 frame of the putative TRGF and 2 stop codons are in the +3 frame,

leaving +1 as the only frame of the putative TRGF free from stop codons conserved across the five species. If I assume that the pseudogene was free from stop codons when it originated, then this implies that the putative TRGF sequence following the repeat region is in the same frame as the original pseudogene sequence. The pseudogene is only free from stop codons between the repeat region and the final stop codon. Given that the length of the pseudogene is not a multiple of three implies that not only substitutions but also indels occurred in this genome region since the origination of the pseudogene. It is not known whether the homologous intact gene has been lost or simply is not present in the genome assembly. The fact that I was unable to identify the ancestral gene associated with this pseudogene, the high level of sequence conservation (95% sequence identity between *D. yakuba* and *D. simulans*, excluding 25 out of 587 nucleotides corresponding to indels), and more than 110 amino acid long ORF still present in *D. yakuba* all add up to substantial evidence that this *D. yakuba* ORF annotated as a pseudogene might in fact be a miss-annotated functional gene. Regardless of whether this pseudogene is a true remnant of a previously functional gene or a miss-annotated gene that is still functional today, I conclude that the putative TRGF did not evolve *de novo*. Instead it evolved from an ancestral protein encompassing all yellow regions shown in Figure 2.6 by truncation of the N-terminal.

2.3 Discussion

The aim of this study was to identify high confidence TRGFs as most promising candidates for experimental studies of protein-coding genes that emerged in the past 8 Mya, while avoiding ascertainment biases associated with preconceptions of how *de novo* genes are born. I will only learn about how different *de novo* genes are from well-established genes if I look for them with an open mind and without assumptions that their sequences must be similar to well-established genes in order to be functional. Unlike other studies, I did not filter out genes with composition distinct from average composition of sequences in protein databases [Vakirlis et al., 2017] nor assume that TRGFs cannot contain splicing signals [Knowles and McLysaght, 2009]. To avoid including candidates that are not functional protein-coding genes, without making such assumptions, I used ORF conservation as a proxy for selection and hence evidence for functionality, in addition to using NCBI genome annotations as the most comprehensive synthesis of evidence for transcription and/or translation. To avoid including candidates that were not born *de novo*, I conducted extensive investigation of homologous non-coding sequences in sister species.

I identified a single TRGF with annotated single copy genes in *D. simulans* and *D. sechellia*. This TRGF is located in a syntenic context conserved across all species in *Drosophila* subgenus. It contains an intron that pre-dates birth as an ORF. I identified potential homologous DNA regions in four additional outgroup species (*D. ananassae*, *D. suzukii*, *D. pseudoobscura* and *D. miranda*), but the sequence

conservation level was insufficient to establish the most likely ancestral state. No start codon was present in these homologous DNA regions. We can thus rule out the possibility that two independent pseudogenization events, one in *D. melanogaster* and one in the basal lineage of the *D. yakuba-erecta* clade, created the illusion of a TRGF as a false positive. Our results highlight that *de novo* gene studies should under no circumstances exclude candidate TRGs just because they have introns.

The number of *de novo* genes reported in any study depends on the balance of false positives and false negatives that has been achieved by the authors. This is shaped by decisions as to what counts as an evidence for functionality and what properties of the candidate genes signify that they are not true *de novo* genes. When I began this study, our requirement that a *de novo* gene must have homologous non-coding DNA sequence(s) in out-group species as evidence for the time of emergence was stricter than most. Since then two papers have been published that described [Vakirlis and McLysaght, 2018] and applied [Zhang et al., 2019b] a similar requirement for homologous non-coding DNA sequences in outgroup species.

Zhang et al. [2019b] examined *de novo* genes in the *Oryza* clade and concluded that about 51.5 *de novo* genes per million years are generated and retained in this clade. While care was taken to show *de novo* status, this number is nevertheless likely inflated by lenient criteria for functionality. Intact gene structure and some transcription and translation were considered sufficient, with no requirement for functional evidence or evolutionary conservation. The estimated rate of *de novo*

gene birth is also potentially deflated (but not by as much) by the assumption that recent *de novo* genes cannot be present in more than one copy. Another limitation of the study is that only the single best hit to a genome was considered. Since hits were accepted if they covered $\geq$20% of an ORF, this could lead to selecting a short highly similar region (for example, to a low complexity region) and ignoring a longer truly homologous region with a slightly lower match score. Accepting matches that cover as little as 20% of an ORF is contradictory to the idea presented in the paper that indels and substitutions are the main ORF triggers, and may have deflated the estimate.

In contrast, our study, which was designed to identify high-confidence experimental candidates, is likely an underestimate, in part because homologous sequence in orthologs might be missing or unrecognisable, but mostly because it cannot find a TRGF unless it is already present in the NCBI gene annotations of two species. The incomplete nature of genome annotation is more of a problem when a gene must be annotated in two species than when it must merely be annotated in one. Abascal et al. [2018] shows that about 12% of human genes have different annotations across the three most popular databases (RefSeq, Ensembl/GENCODE and UniProtKB), and that some genes that are listed as non-coding actually have more experimental evidence for producing a protein than some genes listed as protein-coding. Even in relatively simple species like *Escherichia coli* about 35% of the annotated genes lack experimental evidence of function [Ghatak et al., 2019]. The annotation quality for the *Drosophila* subgenus is unlikely to be better than for the

human genome. Nevertheless, I believe that synthesis of evidence from all data sets submitted to NCBI is by far better than the evidence that I could have gathered and synthesised myself without performing experimental work.

The availability of evidence for functionality is the limiting factor in identifying very young genes. Without it, short young proteins are often left out of genome annotations, and hence alternative approaches like screening all ORFs present in a genome [Ruiz-Orera et al., 2018] are required to identify them. Given the frequency of premature stop codon mutations, conservation of an ORF across several species can be used as a proxy for functionality, as I do here. However, sufficiently short ORFs can still be conserved by chance sometimes across several species.

One reason I find a lower rate of *de novo* gene birth might be that false positive evidence of functionality inflates single-species estimates in other studies, whereas false negative failure to reproduce such evidence in two species deflates it in our study. However, it is also possible that both estimates are approximately correct, with the discrepancy arising from the fact that rapid emergence of functional ORFs is counter-balanced by rapid loss, as discussed by Schlötterer [2015]. Since new-born proteins are not yet integrated in the protein interaction network, they might be relatively dispensable; even if adaptive at first, they might not remain adaptive as the environmental and genetic context changes. In this case our approach, in using evolutionary conservation to exclude non-functional polypeptides, also excludes functional proteins whose functionality is short-lived.

There have been several previous papers aimed at identifying TRGs in *Drosophila* subgenus: the pioneering work of Levine et al. [2006] focused on *de novo* genes, followed by a survey of all TRGs [Zhou et al., 2008], a study about essentiality of TRGs [Chen et al., 2010], an in-depth analysis of the evolution and function of six candidate *de novo* genes [Reinhardt et al., 2013], and a study of very young *de novo* genes in *D. melanogaster* that are still segregating in the population [Zhao et al., 2014]. These studies collectively reported 16 *de novo* protein-coding genes and two *de novo* ncRNAs (Dme_CR32582, Dmel_CR32690) that are fixed in *D. melanogaster* genome and not present outside of the *Drosophila* subgenus. Three of these protein-coding genes (Dmel_CG33235, Dmel_CG33666, Dmel_CG34434) are present only in *D. melanogaster* and hence were not included in our analysis, and another seven of them (Dmel_CG2042, Dmel_CG32582, Dmel_CG32690, Dmel_CG32824, Dmel_CG40384, Dmel_CG9284, Dmel_CG32582) have been removed from the genome annotations since the time of publication. For the remaining six of previously reported *de novo* protein-coding genes, I was able to identify homologous genes outside the *Drosophila* subgenus (Dmel_CG31882, Dmel_CG30395, Dmel_CG31406, Dmel_CG32712) or I was unable to identify homologous DNA regions in any of the outgroup species (Dmel_CG15323, Dmel_CG31909). Note that these last two could still be *de novo* genes. Here I have identified a TRGF containing *Dsim_GD19764* and *Dsec_GM10790* in *D. simulans* and *D. sechellia* respectively that evolved *de novo*. This TRGF is not present in *D. melanogaster* and hence was not part of these previous studies. I did not iden-

tify any TRGFs in this clade that evolved *de novo* and contain an annotated *D. melanogaster* gene. Similarly to previous studies that showed that *de novo* genes in *Drosophila* subgenus are expressed predominantly in testes Levine et al. [2006], Reinhardt et al. [2013], Zhao et al. [2014], the TRGF identified in this work also has transcriptomic evidence of being expressed in testes.

Our results show that while *de novo* genes that are conserved across several species undoubtedly do exist, their number is probably on the lower side of the spectrum of estimates reported in previous studies. I have identified only a single TRGF in the *Drosophila* subgenus, which does not allow me to identify a common pattern of emergence of *de novo* genes. High confidence in its annotation as *de novo* and as conserved may make this *de novo* gene the best candidates in *Drosophila* subgenus identified so far for the experimental studies needed to drive the field forward. To advance our understanding of *de novo* TRGs experimental studies are required. For example experimental studies of *de novo* gene BSC4 which is present only in *Saccharomyces cerevisiae* showed that this gene has synthetic lethal knockouts [Cai et al., 2008] and it has a characteristic three-dimensional fold [Bungard et al., 2017]. Apart from essentiality-testing knockout studies and studies aimed at determining the three dimensional structure of the protein, *de novo* gene field would also benefit from spacial proteomics studies (identifying where in a cell is the protein located), single cell transcriptomic studies (identifying expression patterns relative to other genes expressed in a same cell) and phenotypic studies aimed at elucidating the function of a *de novo* gene.

Chapter 3

Analysing single cell RNA-seq data with generative neural networks

This chapter demonstrates the utility of generative neural networks (GNNs) for analysing single cell RNA sequencing (scRNA-seq) data. Starting from an intuitive concept of Principal Component Analysis (PCA), that provides an idea about variability present in the data, I build up step-by-step to a model based on GNN that can be interpreted in a biologically meaningful way. I focus on what information one can expect to be contained in scRNA-seq data and aim to find an appropriate model architecture to maximise the proportion of the information that can be extracted. The contributions made in this chapter are as follows:

- A novel approach to scRNA-seq data analysis using GNNs
- A thorough assessment of the information flow through an autoencoder
- A comprehensive analysis of a latent space embedding created by an autoencoder

3.1 Materials and methods

3.1.1 Dataset processing

To demonstrate the properties of scRNA-seq data and the capabilities of GNNs I will use the human skin dataset produced by Cheng et al. [2018]. I chose this dataset for several reasons. First, this dataset contains a large amount of cells from a specific tissue (human skin) which makes it suitable for a GNN training. Second, due to the experimental setup and the properties of the skin tissue, the dataset captures many different aspects of scRNA-seq data - batch and patient effect, cell cycle effect, cells from healthy and inflamed tissue, cell types that are related to each other through differentiation, cell types that are not directly related to the rest of the sample. Finally, another human skin dataset produced by Tabib et al. [2018] is available, which will allow me to test whether a GNN trained on one dataset can be applied to a dataset produced in a different lab using a different protocol.

The dataset produced by Cheng et al. [2018] contains 92889 human epidermal cells from 12 samples. The samples were normal surgical tissue discards from three different anatomic sites - adult scalp epidermis, adult truncal epidermis (healthy and psoriatic) and neonatal foreskin. The single cells were dissociated and then put through fluorescence-activated cell sorting (FACS) to exclude dead cells, doublets, and debris. Libraries were prepared using Chromium Single Cell 3' v2 protocol from 10X Genomics and sequenced with either Illumina HiSeq 2500/4000 or NovaSeq 6000.

Data quality control was already performed by Cheng et al. [2018]. They filtered out cells with lowest and highest number of detected genes (0.5% and 15% of the data respectively) to remove empty droplets and doublets respectively. Usually the number of UMIs not the number of detected genes is used to identify doublets. To confirm the absence of doublets in the dataset, I assessed the distribution of number of UMIs per cell. A high proportion of UMIs associated with mitochondrial genes is indicative of cells that are either stressed (upregulated mitochondrial gene expression) or damaged (cytoplasmic mRNAs are lost through a broken cell membrane but mRNAs inside mitochondria are retained) [Ilicic et al., 2016]. The authors filtered out cells with highest proportion of mitochondrial gene counts (5% of the data). A typical scRNA-seq data analysis pipeline includes identification of genes with high information content (measured by coefficient of variation, mean adjusted variance or a similar metric), the genes with low information content are then removed from the dataset. In contrast, I treat scRNA-seq data as compositional data (because it is compositional data, even though it is widely ignored) and hence I do not filter genes by their information content simply because such information is not available. See Section 1.2.2.3 for an explanation about why gene expression values are not comparable across cells. Unlike most of the analysis pipelines, I identify and remove cells with low information content. I used coefficient of variation (the ratio of the standard deviation to the mean) as a proxy for information content. I calculated these using only non-zero expression values to limit the effect of technical dropouts - genes that were expressed but not captured due to chance. I

assessed the distribution of the resulting coefficient of variation and removed 0.5% of the data with the lowest information content. Once the quality control has been performed I did not return to this step later in the project, contrary to the suggestion from Luecken and Theis [2019] to first investigate the effects of quality control stringency on the analysis and then adjust the quality control procedure as necessary.

The ability of a neural network to learn from a dataset and the achievable level of performance depends on the representation of the data. Finding an appropriate representation is hence a single most important step in creating a neural network-based model. There are two aspects of scRNA-seq data that can hinder the neural network's ability to learn, but can be alleviated with a data transformation. First, scRNA-seq data is overdispersed - as the true expression level of a gene increases the variability of the expression estimate grows unboundedly. Applying $\log_2(x+1)$ log-transformation shrinks the differences between large values in the dataset and thus stabilises the variance while still preserving their rank order within each cell. Second, the maximum expression value in the cell is not informative. It is likely that the gene with the highest expression value is amongst the real most highly expressed genes, but the exact number of transcripts captured is due to chance. If more transcripts of other genes are captured then the number of transcripts corresponding to the most highly expressed gene will be lower. Luecken and Theis [2019] advocated the log-transformation of the data for two reasons - because it partially mitigates the meanvariance relationship in the data, and because distances between logtransformed expression values represent log-fold changes, which are

the canonical way to measure changes in expression. To avoid the maximum expression value in each cell being misinterpreted by a neural network as genuine differences between the cells I scaled the data to [0,1] range. Both of these data transformations are usually included in scRNA-seq data analysis pipelines regardless of the downstream methods used. An alternative line of reasoning leads to the same protocol. Since neural networks cannot be trained on discrete data, it is necessary to convert it to continuous data. Scaling the data to a fixed range is a good option, but it is sensitive to outlier values - a single large expression value will force the rest of the expression profile to be compressed into a narrow range of values that is detrimental to the training process. Log-transforming the data prior to scaling alleviates that.

To visualise scRNA-seq data I used t-distributed stochastic neighbor embedding (tSNE) implementation by Pedregosa et al. [2011]. tSNE algorithm first creates a probability distribution using the Gaussian distribution that defines the relationships between the points in high-dimensional space. It then uses the Student t-distribution to recreate the probability distribution in low-dimensional space. The heavier tails of Student t-distribution prevent points from getting crowded in low-dimensional space due to the curse of dimensionality. tSNE creates a low-dimensional projection of the data based on the local relationships between data points, thus capturing the non-linear structure of the data. Starting with a random embedding, t-SNE optimizes the embedding using a gradient descent. For this reason no “mapping” is created, i.e. it is impossible to add more data points later to the same embedding.

The process of embedding optimisation is governed by two important parameters. The perplexity parameter determines how many neighbouring points are taken into account when determining a position of a given point. Setting this parameter to a value smaller than the number of points in the dataset prevents any single point from having a disproportionate influence on the whole embedding, i.e. makes the algorithm robust to outliers. The learning rate parameter determines the dynamics of the parameter value optimisation. At the two extremes, a very low learning rate results in most points being compressed in a dense cloud with few outliers floating around and a very high learning rate results in the data looking like a ball with any point approximately equidistant from its nearest neighbours. tSNE algorithm does not involve a clustering step, it learns from the data and generates a low-dimensional embedding of the data. It is up to a user to infer clusters from the resulting low-dimensional embedding. I used random initiation, learning rate equal to 500 and 1000 iterations for training. The perplexity parameter was set to 50, corresponding to the smallest cell cluster that could be of interest in the intended downstream analysis.

3.1.2 Autoencoders

Generative neural networks are feedforward neural networks that learn through data reconstruction. Autoencoders are the simplest type of generative neural networks, they make minimal assumptions about the data. An autoencoder consists of two parts - the encoder that reduces dimensionality of the data and the decoder that

reconstructs the data from the lower dimensional embedding back to original dimensionality. The lower dimensional embedding is called the latent space. Figure 3.1 shows the usual ball and stick diagram of an autoencoder - it emphasises the values in the input data (the top of the diagram), the fully connected layers that pass the information through the network all the way down to the reconstructed values (the bottom of the diagram). Figure 3.2 shows a more useful representation of an autoencoder that illustrates the linear algebra underpinning the network. The input data, the lower dimensional representations of the data and the reconstruction of the data are shown on the left. The components of an autoencoder are shown on the right. There are four layers in this autoencoder. A layer of an autoencoder, or indeed any neural network, consists of a matrix of weights and a vector of biases. The dimensionality of the matrix matches that of an input and the desired dimensionality of the output. I will use a gene expression profile as a concrete example of an input data. The gene expression profile (a matrix with 1 row and the number of columns corresponding to the number of genes) is an input into the first layer of the autoencoder. The input is first combined with the matrix of weights via matrix multiplication, and then the biases are added. In Figure 3.2 the biases are not shown explicitly to avoid clutter, adding them is implied in the arrows. The dimension of a matrix that is not constrained by the properties of the input is often referred to as a number of nodes in the layer. The output of a layer contains as many features as there are nodes in this layer. Different types of layers are used for different applications, in the case of scRNA-seq data I will only use fully connected layers, which means that every value in the input is multiplied with every value in the correspond-

ing raw of the weight matrix. The layer with the lowest dimensionality is called the latent layer. The output of the latent layer is called the latent representation of the data. The training of the network happens via backpropagation - the information is travelling back from the output of the final layer of the autoencoder in the direction of the input data, i.e. the arrows in the diagram are reversed. A "cycle" of training is complete when all of the training data passes through the network. This is called an epoch. The training process consists of hundreds of epochs, i.e. the whole dataset is used repeatedly to train the network. Hereafter I will refer to "upstream" and "downstream" in the network meaning the direction of the data passing through the network (coinciding with the arrows). For the autoencoder implementation used in this and the following chapter I employed weights and biases initialisation with uniform distribution as described by [He et al., 2015a] and a popular optimisation algorithm Adam [Kingma and Ba, 2014] with default parameters.

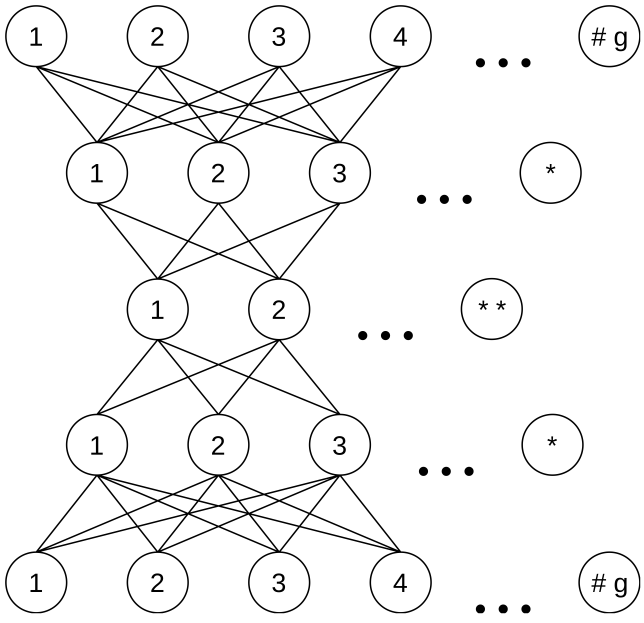


Figure 3.1: A ball and stick diagram of an autoencoder.

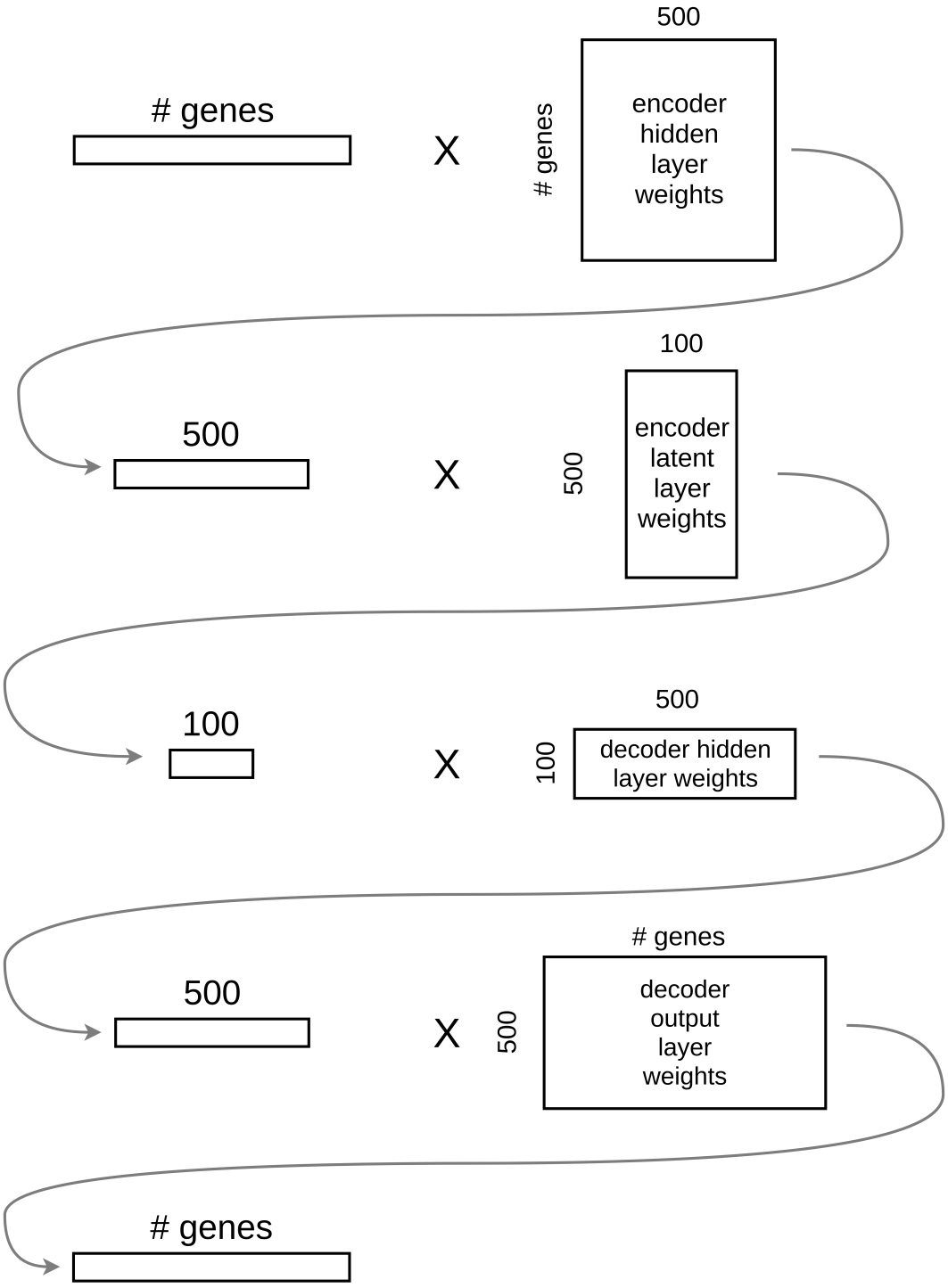


Figure 3.2: A deep autoencoder.

3.1.3 Technical implementation

Python (v3.7.0) was used to create all the models described in this chapter and perform all the experiments described. This work heavily relies on scipy (v1.1.0) [Jones et al., 2001], numpy (v1.15.2) [van der Walt et al., 2011], scikit-learn (v0.20.0) [Pedregosa et al., 2011] and pandas (v0.23.4) [McKinney, 2010]. All GNN-based models were created using PyTorch (v.1.1.0.post2) [Paszke et al., 2017]. Learning process was monitored using tensorboardX (v1.5). All plots were created using matplotlib (v.3.0.0) [Hunter, 2007].

3.2 Results

3.2.1 Properties of single cell RNA-seq data

To demonstrate the properties of scRNA-seq data and the capabilities of GNNs I will use the human skin dataset produced by Cheng et al. [2018]. The dataset contains 92889 cells from 12 samples. Data quality control was performed by the authors. Figure 3.3 shows the relationships between the number of different genes detected per cell, the number of unique transcript (UMIs) detected and the proportion of the transcripts corresponding to mitochondrial genes. A high proportion of UMIs associated with mitochondrial genes is indicative of cells that are either stressed (upregulated mitochondrial gene expression) or damaged (cytoplasmic mRNAs are lost through a broken cell membrane but mRNAs inside mitochondria are retained) [Ilicic et al., 2016]. Figure 3.3 shows that all cells have less than 10% of transcripts corresponding to mitochondrial genes. I removed a single cell with

the number of unique transcript 42% higher than the cell with the second highest number due to a high chance of it being a doublet that escaped FACS. To assess the information content in cell expression profiles, I examined the relationship between the coefficient of variation, the total number of expressed genes and the maximum expression value, see Figure 3.4a. It is clear that the higher numbers of different genes detected correspond to the higher maximum expression values. As expected, the coefficient of variation, which I used as a proxi for information content, increases with the maximum expression values. The cells with very low maximum expression values, see Figure 3.4b, also have very small numbers of genes detected. This is indicative of a bad quality cells due to either biological or technical limitations. I removed 0.5% of the cells with the lowest information content, corresponding to the coefficient of variation cut-off value of 2. This also removed the cells with maximum gene expression value below 50 unique transcripts per gene. After this additional quality control, the cleaned dataset contained 92423 cells, see Table 3.1. The number of cells from each sample that were excluded during quality control stage indicates that all the samples are of similar quality. A test set containing 20% of the data selected at random will be used to control for overfitting during GNN training. Table 3.1 shows that the composition of the test set reflects the composition of the whole data set.

To address certain properties of scRNA-seq data, I log-transformed the data and then scaled it to [0,1] range, see Section 3.1.1 for motivation behind this. To estimate the variability of amount of information contained in each expression profile

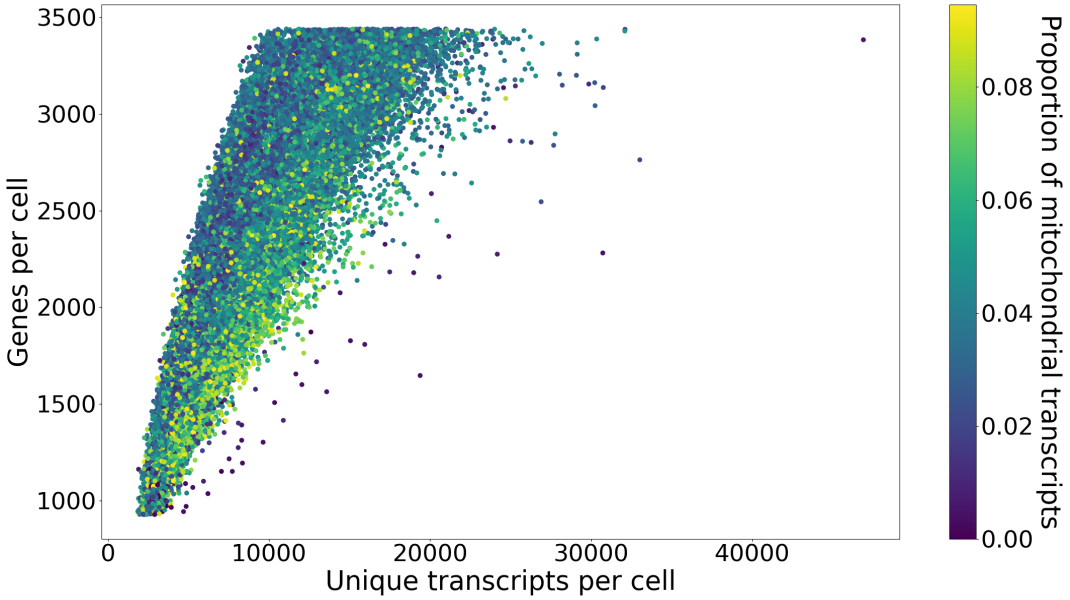
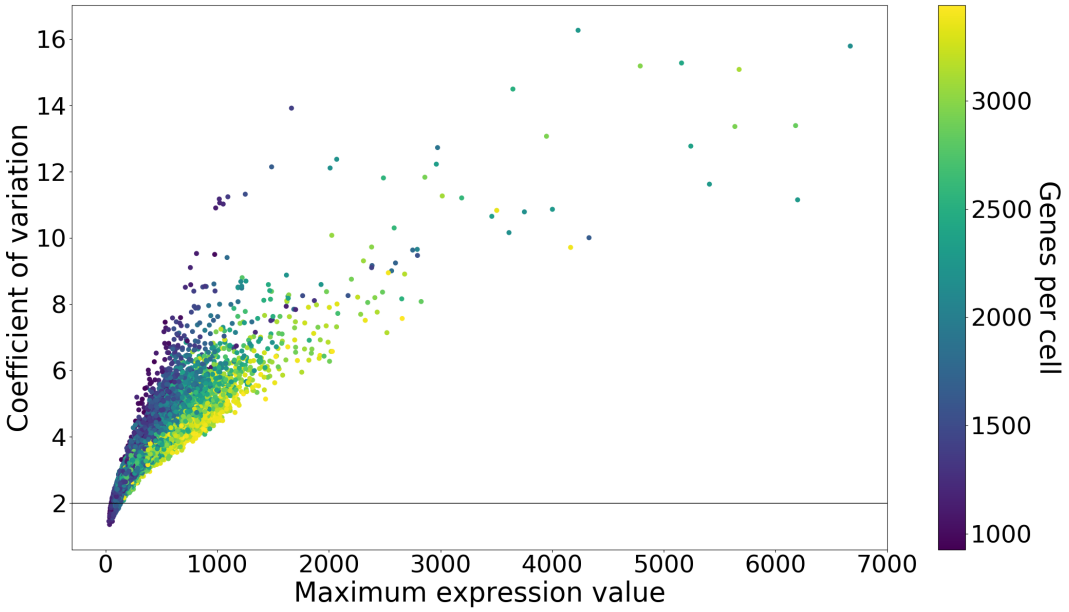


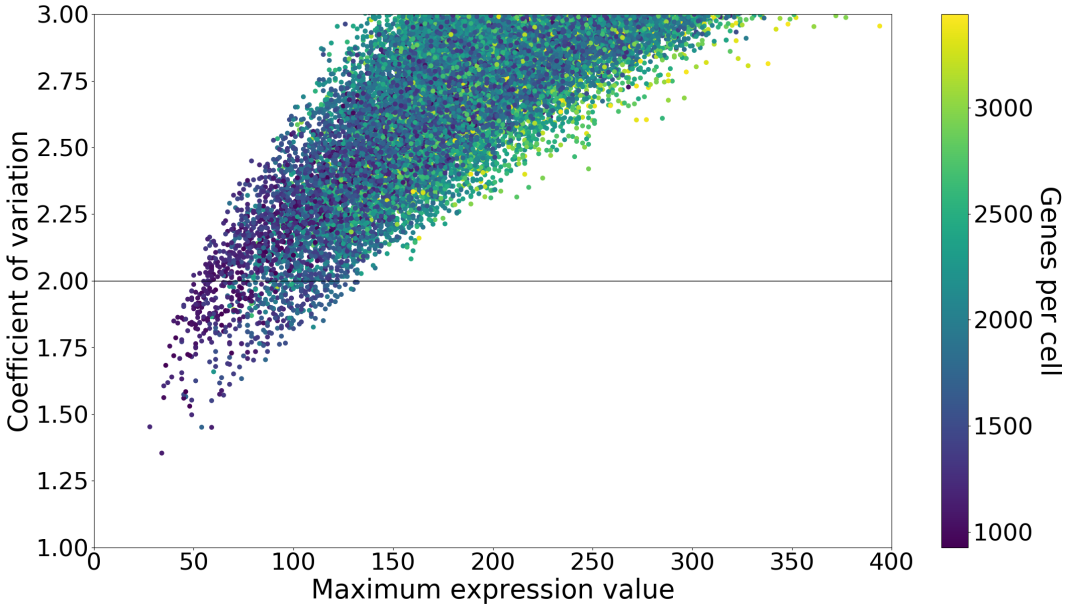
Figure 3.3: The relationships between the number of different genes detected per cell and the number of unique transcripts (UMIs) detected. The cells are coloured by the proportion of mitochondrial transcripts. Number of genes detected per cell ranges from 928 to 3440, number of transcripts per cell range from 1886 to 33015.

Table 3.1: Number of cells per sample

Tissue	Sample ID	Total cells	Training set	Test set	Excluded cells
Scalp	11	2381	1838	476	67
	26	8054	6397	1637	20
	32	10126	8060	1988	78
Trunk	4	12116	9695	2353	68
	41	7104	5694	1387	23
	53	5909	4703	1184	22
Foreskin	8	8030	6367	1596	67
	9	7387	5825	1499	63
	12	10757	8506	2201	50
Psoriasis	14	9743	7769	1973	1
	48	5709	4604	1099	6
	49	5573	4480	1092	1



(a) The whole data.



(b) Cells with coefficient of variation below 3.

Figure 3.4: The relationship between the maximum expression value in a cell and a coefficient of variation (calculated using non-zero expression values only). The cells are coloured by the number of genes detected in a cell.

after the transformation, I examined the distribution of the values corresponding to the two-fold differences in expression, see Figure 3.5. As expected, the distribution is approximately normal with no outliers since cells with low information content have already been removed.

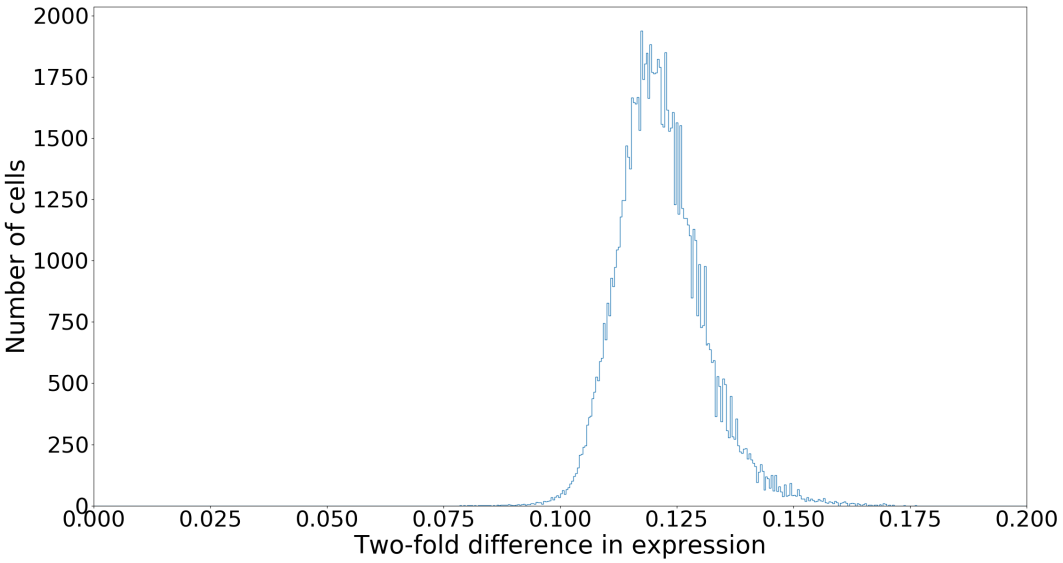


Figure 3.5: The distribution of values in log-transformed data scaled to [0,1] range corresponding to the two-fold differences in expression. The values range from 0.0787 to 0.1763.

Genes that have been captured in a small number of cells do not contain enough information to influence neural network training, but they slow down the training due to their computational footprint. As recommended by Luecken and Theis [2019], I filtered out the genes expressed in a number of cells smaller than the smallest cell cluster that could be of interest in the intended downstream analysis. Informed by the cumulative proportion of the genes present in a certain number of cells, see Figure 3.6, I chose an arbitrary cut-off value of 50. This reduced the total number of genes by 4.4% to a total of 18962. I avoided a common mistake of first removing

uninformative genes and then looking at the number of genes/UMIs per cell and/or transforming data in a manner that assumes that the whole expression profile is taken into account. Figure 3.7 reveals the relationship between the mean and the variance of the expression level of a gene (both calculated using only non-zero expression values) and the proportion of the cells in the dataset in which this gene has been detected. The variances are lower relative to the means in genes expressed across a larger proportion of cells. This plot shows that, even though the expression values of a gene are not comparable across cells, there is a pattern in the expression values.

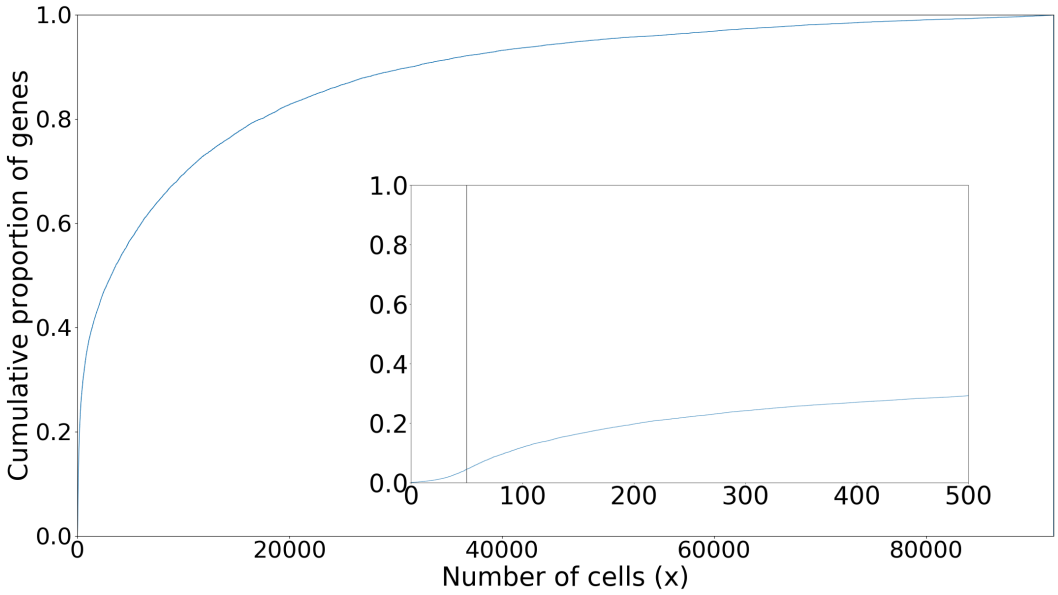


Figure 3.6: The cumulative proportion of genes present in at least x cells. The insert shows the shape of the function on the range x $\in$ [0,500]. The vertical black line in the insert show an arbitrary cut-off value of 50 that has been chosen.

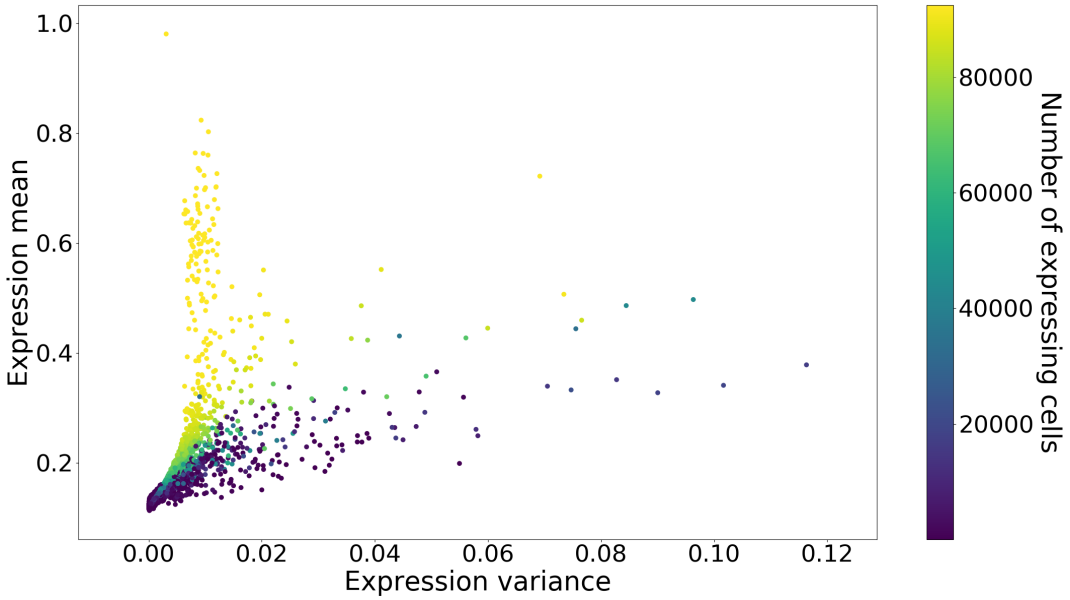


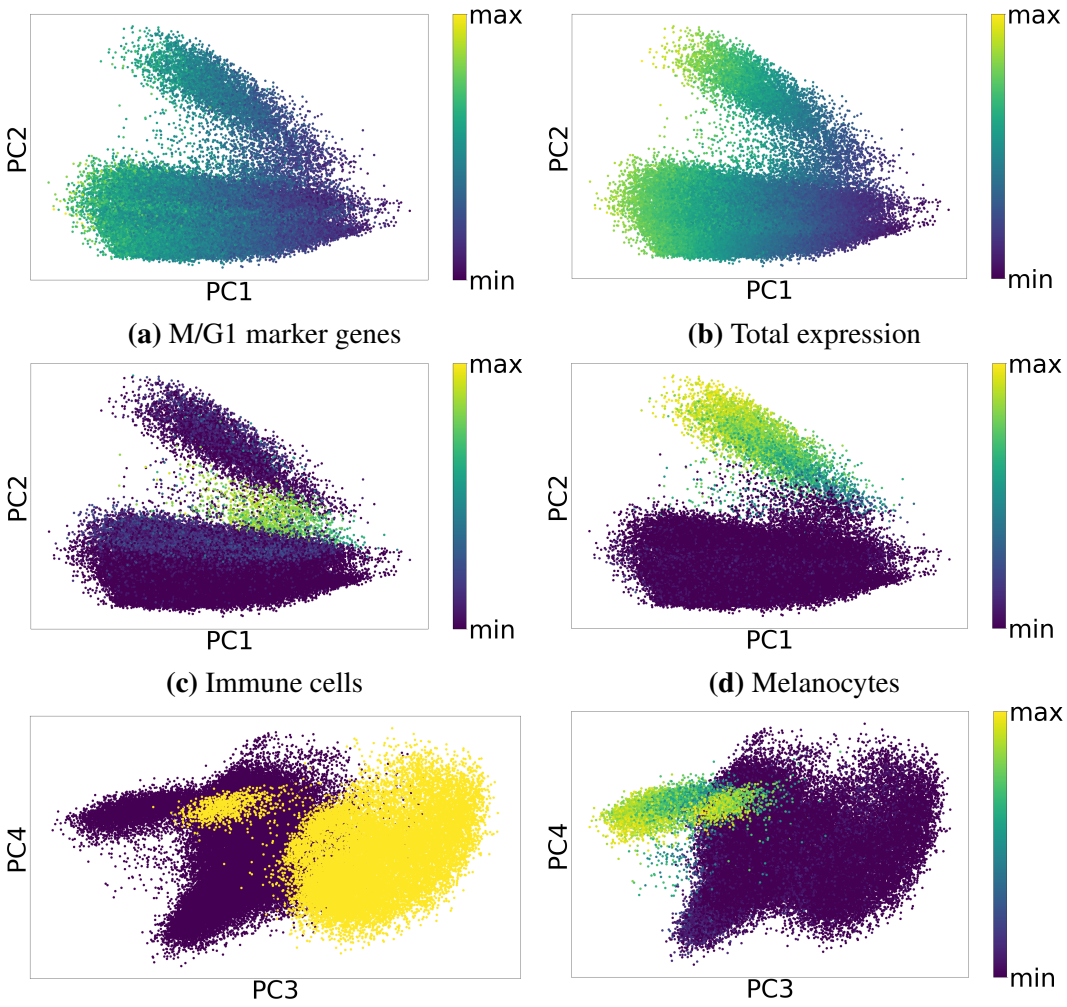
Figure 3.7: The relationship between the mean and the variance of expression values (calculated using non-zero expression values only) for each gene. The genes are coloured by the number of cells in which this gene's expression has been detected.

3.2.2 scRNA-seq data through the lens of PCA and tSNE

The linear combinations of the gene expression values that correspond to the largest variance in the data can be identified with PCA. There are two limitations of PCA to keep in mind. First, this analysis is designed to find principal components (PCs) that explain as much variability in the data as possible. As a result, a couple of leading components will correspond to several biological pathways bundled together with no information about how to disentangle those signals. This lack of biological relevance of PCs has been also noted by [Tan et al., 2016]. Second, PCA assumes orthogonality of important features. Hence, it results in decompositions based solely on *a priori* defined statistical constraints which are not likely to have any relationships to biological pathways. PCA is also sensitive to the relative magnitudes of variables in the dataset, but this is not much of a problem for the data scaled to a

specific range. Highly expressed genes will be upweighted by PCA, which might or might not be biologically meaningful. PCA of the Cheng et al. [2018] dataset identifies 4 PCs with high variances. The first PC corresponds to the expression of M/G1 cell cycle phase marker genes identified by Macosko et al. [2015], which also correlates with the total expression per cell, see Figure 3.8 a-b. The second PC separates immune cells (identified by expression of marker genes CD74 and HLA-DPA1) and melanocytes (identified by expression of marker genes PMEL and TYRP1) from the rest, see Figure 3.8 c-d, and the third separates melanocytes and psoriasis samples from the rest, see Figure 3.8 e-f. The role of the forth PC is not known. The rest of the PCs have relatively small variance and are likely to be meaningful only in combinations. It is more appropriate to use singular value decomposition (SVD) instead of PCA for sparse datasets like scRNA-seq. However, SVD leads to results very similar to Figure 3.8 and in practice provides no advantages in this case.

To be able to compare the performance of PCA to the performance of other more complicated models, a performance metric is required. Since the GNN-based models I will be comparing with learn by reconstructing the input data, I will use the sum of square errors between an input expression profile and the corresponding reconstruction as a performance metric. scRNA-seq data is noisy and it is neither expected nor desirable that the model reconstructs both the signal contained in the data and the random noise. Using the squares of the differences between the real and the reconstructed expression values is therefore appropriate as it penalises large discrepancies more than small discrepancies that are likely to be associated with



(a) M/G1 marker genes **(b)** Total expression

(c) Immune cells **(d)** Melanocytes

(e) Psoriasis samples **(f)** Melanocytes

Figure 3.8: (a) PC1 corresponds to the expression of M/G1 cell cycle phase marker genes. The cells are coloured by the total expression of M/G1 cell cycle phase marker genes. (b) PC1 also corresponds to the total expression per cell. The cells are coloured by the total expression. (c) PC2 separates immune cells. The cells are coloured by the total expression of marker genes CD74 and HLA-DPA1. (d) PC2 separates melanocytes. The cells are coloured by the total expression of marker genes PMEL and TYRP1. (e) PC3 separates psoriasis samples. Psoriasis samples are shown in yellow, other samples in purple. (f) PC3 separates melanocytes. The cells are coloured by the total expression of marker genes PMEL and TYRP1.

noise in the data. The square error is similar to using the epsilon-insensitive error but without the need to choose an arbitrary cut-off value. Hereafter I will refer to the average mean square error per cell as the reconstruction error. Using 100 PCs to reconstruct the dataset results in the reconstruction error equal to 27.6746. To put this into perspective, using 100 most highly expressed genes instead of PCs would lead to the reconstruction error equal to 74.3654.

To visualise the data in a two-dimensional plot most scRNA-seq studies use t-SNE [van der Maaten and Hinton, 2008] that provides a non-parametric mapping to a lower number of dimensions, typically two. The mapping is learned through iterative optimisation-based adjustment and no interpretable information about the mapping is stored. Figure 3.9 shows a tSNE plot produced based on 100 PCs, see Section 3.1.1 for details. Cells from every sample form 2 large per sample. Additionally, there are small clusters containing cells from different samples. Colouring the same plot based on the marker gene expression levels allowed me to identify the cell types corresponding to each of the clusters. I used marker gene expression as a proxy for cell type identification, an approach similar to the one employed by Cheng et al. [2018]. Basal skin cells can be identified by high collagen gene expression, here I used COL17A1 collagen gene. Suprabasal skin cells can be identified by high keratin gene expression, here I used combined expression level of KRT1 and KRT10 keratin genes. Figure 3.10 a-b shows that the two clusters corresponding to each of the samples contain basal and suprabasal cells respectively. Immune cells (identified by combined expression of marker genes CD74 and HLA-DPA1)

from all samples were clustered together into several tight clusters, Figure 3.10 c. Similarly, melanocytes (identified by combined expression of marker genes PMEL and TYRP1) clustered in one part of the plot, Figure 3.10 d. In each cluster cells are arranged along the gradient of the total expression of M/G1 cell cycle phase marker genes, Figure 3.10 e. From the plots in Figure 3.10 I conclude that clusters produced by tSNE algorithm correspond to biological cell types and hence can be meaningfully interpreted. It is, however, not known whether the clusters resulted from unique expression profile features identified by the algorithm or were simply driven by the high expression values of known marker genes. The fact that basal and suprabasal skin cells from each cluster form a separate cluster, as well as the fact that the cells in each cluster are arranged based on the cell cycle phase implies that the data contains both technical (batch effect) and biological (cell cycle effect) noise.

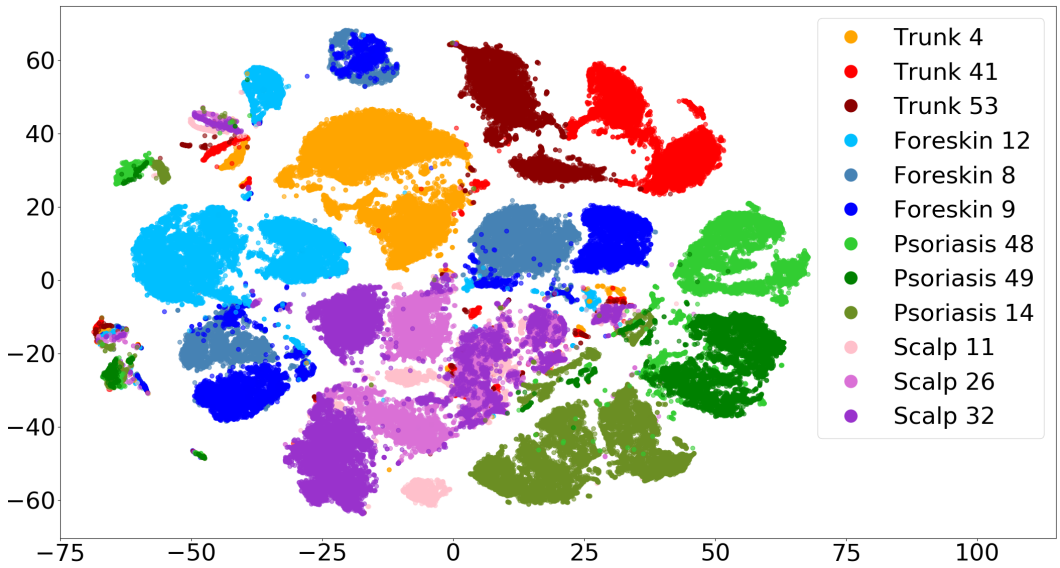
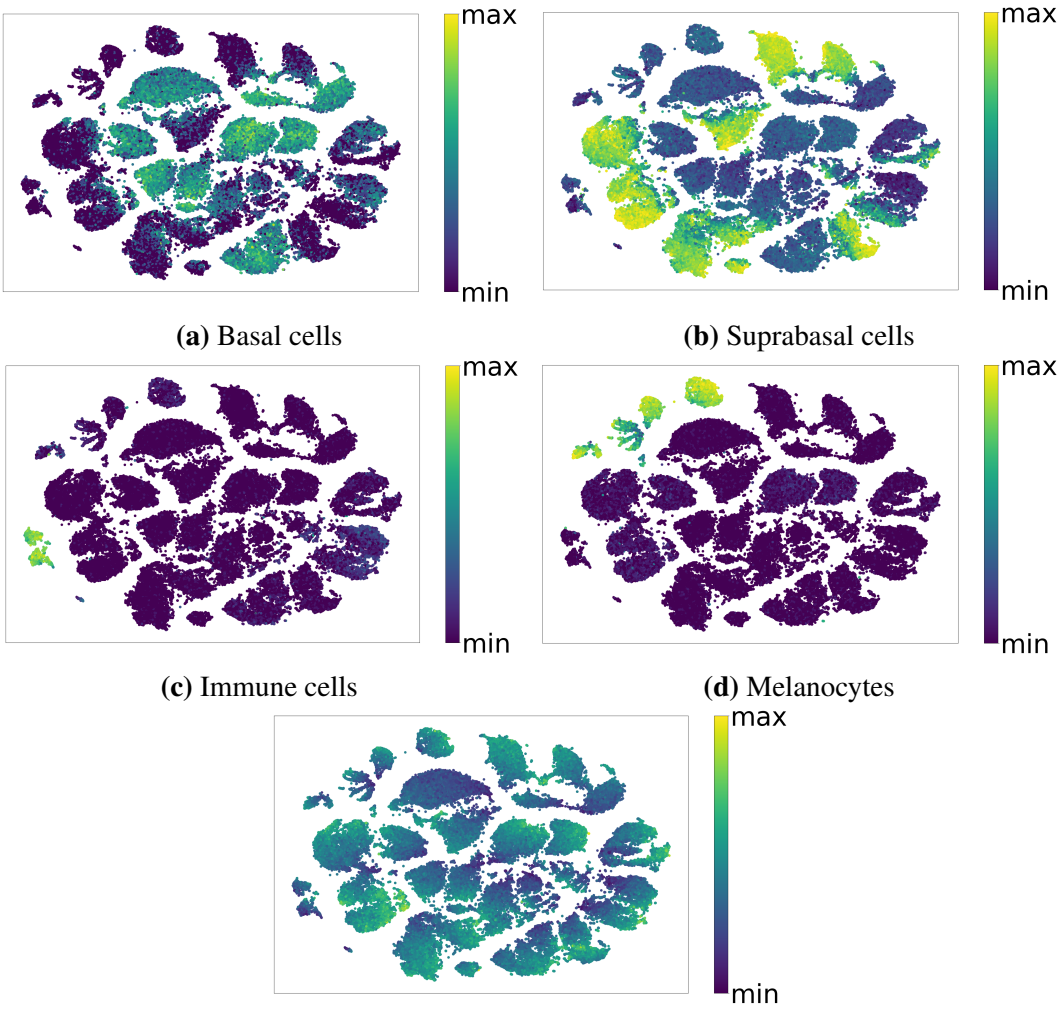


Figure 3.9: tSNE of 100 PCs coloured by sample.



(a) Basal cells

(b) Suprabasal cells

(c) Immune cells

(d) Melanocytes

(e) M/G1 marker genes

Figure 3.10: tSNE of 100 PCs is able to separate (a) basal and (b) suprabasal cells from each sample, (c) immune cells and (d) melanocytes. (e) Cells in these clusters are arranged by the expression level of M/G1 cell cycle phase marker genes. In (a-d) the cells are coloured by the total expression of the corresponding marker genes. In (e) the cells are coloured by the total expression level of M/G1 cell cycle phase marker genes.

3.2.3 From PCA to a deep autoencoder

The two limitations of PCA discussed above can be eliminated by using a shallow linear autoencoder that is mathematically equivalent to PCA. An autoencoder trains by mapping the data into a lower dimensional space and subsequently reconstructing the data from this lower dimensional representation, see Section 3.1.2 for details. The training starts from a random mapping to and from the lower dimensional representation and a very poor reconstruction quality. The training proceeds by iteratively adjusting the mapping guided by the optimiser aimed at minimising the reconstruction error. Since there is no incentive to explain as much variance of the data as possible in a single component, an autoencoder will partition the variance across each component in a random way with the sole goal of producing the best possible reconstruction of the data. Only if the number of components (i.e. the lower number of dimensions to which the data is mapped) is insufficient, there is an incentive to create uncorrelated features. Hence, an autoencoder with a number of components appropriate for the data will produce features that are free from constraints related to variance partitioning and feature orthogonality.

There are two aspects to considered before training an autoencoder - the number of components and the termination criterion for the training. Unlike in PCA where a desired number of components can be selected based on the total proportion of the variance they explain and the incremental gain from adding another component, the decision about the number of components in an autoencoder has to be made *a priori*. Yehudai and Shamir [2019] showed that having an over-parameterised

autoencoder, i.e. the number of components larger than what is required to achieve a good reconstruction quality, usually has no negative effect on the performance as extra components will simply remain unused. Hence the only danger is selecting a number of components that is too low. The upper bound for the number of components is the number of features in the data (in this case, 18962 genes). An autoencoder with 18962 or more components will approach an identity matrix as it trains. Since there are about 1500 TFs in a human genome [Garcia-Alonso et al., 2019] and not all of them are active in an adult skin tissue, it is reasonable to assume that the number of components should be less than that. In practice, the quality of the data determines the maximum number of meaningful components that could be obtained from it. Previous studies compared the performance of GNN-based models with different number of components. Instead I arbitrarily set the number of components to 100 exploiting the fact that the presence of unused components in a trained model is indicative of the number of components being too high.

The training of an autoencoder is governed by a loss function that measures the reconstruction quality, see Section 3.2.2 for the description of the loss function used. The training will proceed indefinitely unless a termination criterion for the training is specified. The aim is to identify useful features present in the data without "memorising" the data. Arpit et al. [2017] showed that, while neural networks are capable of memorising noise, they tend to prioritise learning simple patterns first. This implies that an optimisation strategy, when employed properly, allows a neural network to take advantage of patterns in the data even if the capacity of the

network is sufficient to memorise the data. Arpit et al. [2017] showed that a general analysis of the effective capacity of a neural network is unlikely to be successful since the balance between learning the patterns in the data and memorising the data depends largely on the training data itself. Neto [2018] used their novel approach for differentiating between memorisation and learning to confirm previously reported surprising results that a neural network trained on Gaussian noise is able to learn, not only simply memorise the noise. Previous studies applying GNNs to scRNA-seq data compared the performance of models trained for a different number of epochs, i.e. using different number of times the whole dataset is used to train the network. Instead, I used the value of the loss function measured on the test dataset to detect overfitting. As the training begins, the loss function measured on the test dataset decreases monotonically as the model learns the patterns that generalise to the whole dataset. The loss function for the batches of training data oscillates but exhibits a general downward trend. After a certain number of epochs, the model starts to memorise the training data which no longer generalises to the whole dataset, and hence the loss function measured on the test dataset starts to increase monotonically, see Figure 3.17a for an example. This simple trick allows me to train the model to optimality without including the features that don't generalise beyond the immediate training data.

A shallow linear autoencoder with 100 components (which I will refer to hereafter as nodes) in the latent layer trained to optimality as described above is able to reconstruct the dataset with the reconstruction error equal to 27.6929. This is comparable

to PCA performance. Similarly to PCA, this model learns linear combinations of the gene expression values and reconstructed gene expression values can be negative. To ensure that the reconstructed expression values lie in the same range as the original data I used the Sigmoid activation function

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

for the output layer, thus transforming the output to the (0,1) range. This results in a improvement - the reconstruction error equal to 27.5816. While this model is no longer equivalent to PCA, it does share the main limitation - it assumes that the relationships between the genes is constant throughout the dataset. To create a model capable of capturing the relationships between the genes even if they differ among subsets of a dataset, at least one hidden layer must be introduced.

Adding a hidden layer with a non-linear activation function (consecutive linear layers are mathematically equivalent to having a single linear layer) to both the encoder and the decoder results in two major benefits. First, it removes the assumption that the relationships between the genes is constant throughout the dataset. Second, it captures non-linear interactions between the genes. Two decisions about the network architecture have to be made at this point - the number of nodes in the hidden layers and the activation functions used. As discussed in Section 3.2.3, the number of nodes larger than required usually has no negative effect on performance as extra components will simply remain unused. Hence, I arbitrarily set the number

of nodes in hidden layers to 500. Among activation functions, the Rectified Linear Unit (ReLU) activation function

$$\mathrm{ReLU}(x) = \max(0, x)$$

is the most widely used in deep neural networks due to its attractive properties. Since it is impossible to know in advance which activation functions are most suitable for a particular application of GNNs, I will explore combinations of different activation functions using ReLU as a benchmark.

First, I assessed the performance of the non-linear autoencoder with 500 nodes and ReLU activation in the encoder hidden layer, 100 nodes and ReLU activation in the latent layer and a symmetrical decoder with Sigmoid activation in the output layer. It took 90 epochs to train this autoencoder to optimality before the onset of overfitting. To assess whether the whole capacity of the neural network has been utilised, I plotted the weights and the biases of the encoder's hidden layer, see Figure 3.11. At the beginning of the training both the weights and the biases are initiated with values close to 0, then during the training these values are incrementally adjusted to improve the performance. Figure 3.11 shows that 270 (on the right hand side of the plot) out of 500 nodes have a wide range of weights associated with them and a non-zero bias. In contrast, 230 nodes (on the left hand side of the plot) remained "unused" as all of the weights associated with them remained close to their initial value, i.e. did not change during training. Similarly, the weights

and the biases of the latent layer show that 64 of 100 nodes have been used. This implies that increasing the number of neurons in the network will not improve its performance. The performance of this autoencoder is comparable to PCA - the reconstruction error is equal to 27.6932. The reason for the lack of improvement becomes apparent when examining the embedding of the data in the latent space. Figure 3.12 shows the distribution of 92889 data points in each of the 64 "used" dimensions. All of these 64 distributions are zero-inflated, which is the result of using ReLU activation function - $\mathrm{ReLU}(x) = \max(0, x)$. This is an unsuitable latent space shape, and forcing the data embedding into it results in poor reconstruction quality. It is apparent that choosing a more suitable activation function can improve the performance of the autoencoder.

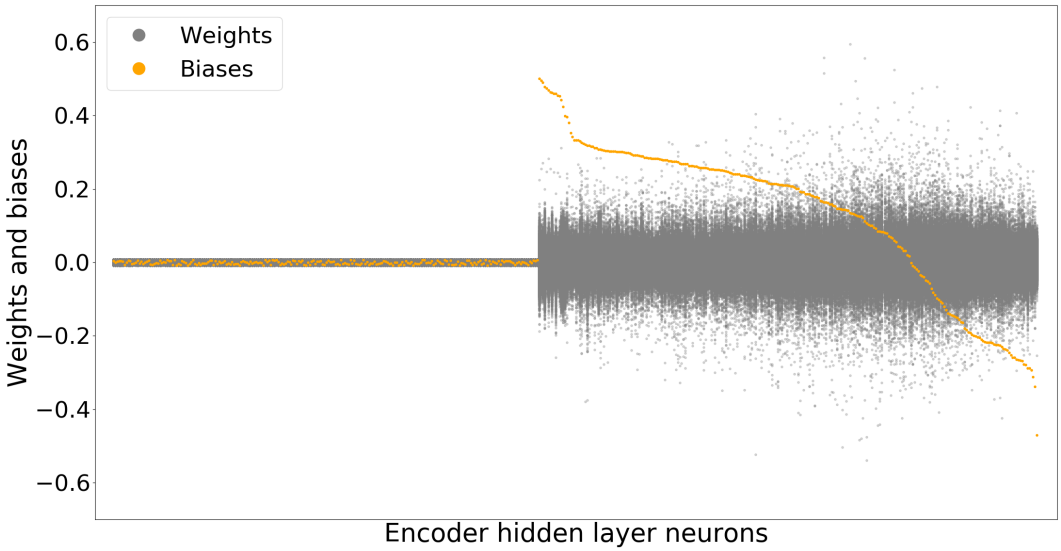


Figure 3.11: A trained non-linear autoencoder with 500 nodes and ReLU activation in the encoder hidden layer, 100 nodes and ReLU activation in the latent layer and a symmetrical decoder with Sigmoid activation in the output layer. The plot shows weights and biases of nodes in the hidden layer of the encoder. The nodes are arranged as follows (from left to right): 230 "unused" nodes followed by 270 "used" nodes ordered by the value of the bias associated with them.

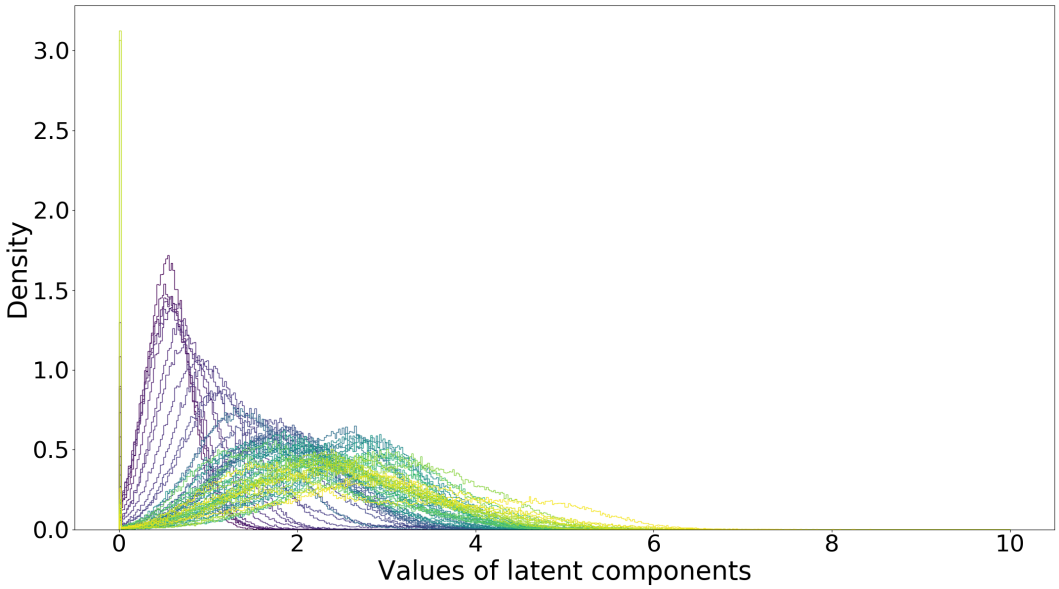


Figure 3.12: A trained non-linear autoencoder with 500 nodes and ReLU activation in the encoder hidden layer, 100 nodes and ReLU activation in the latent layer and a symmetrical decoder with Sigmoid activation in the output layer. The plot shows distributions of 64 latent components with the highest variance. Each of the components is shown in a different colour simply for ease of visualisation.

A choice of an activation function determines the distribution of values in the output of that particular layer. Critically, a choice of a sequence of activation functions in the layers of a deep neural network determines the ability of the network to learn. For the latent layer it is important to have an activation function that produces distributions of values in latent components that are both suitable for capturing features of a particular type of data and easy to sample from. For scRNA-seq data it is important to be able to produce non-unimodal distributions corresponding to features such as differentiation status - most cells in the sample will be either pluripotent or differentiated, while differentiating cells will be in the minority. Bounded distributions allow for simple uniform sampling. The ability of the model to capture biologically meaningful features equally depends on the interplay between the activation functions used in hidden, latent and output layers. I used the same activation

function for both the encoder's and the decoder's hidden layer.

To identify the best pair of the latent and hidden layer activation functions I tested all combinations of the following potentially suitable activation functions:

$$\mathrm{ELU}(x) = \max(0,x) + \min(0, e^x - 1),$$

$$\mathrm{SoftSign}(x) = \frac{x}{1+|x|}, \quad \mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

Sigmoid and ReLU. Autoencoders with Sigmoid or Tanh activation in the latent layer and one of the following activations in the hidden layers - ELU, SoftSign, Tanh or ReLU - were not able to train on this data. See Table 3.2 for the performance comparison of autoencoders that trained successfully, the architectures that were not able to train are not presented in this table. The reconstruction error decreases as the number of used components increases, see Figure 3.13. The distributions of values in the latent components produced by each autoencoder are either unimodal only, both unimodal and bimodal, or include distributions with one, two and more modes.

Based on the analysis above, I chose the non-linear autoencoder with 500 nodes and ELU activation in the encoder hidden layer, 100 nodes and Softsign activation in the latent layer and a symmetrical decoder with Sigmoid activation in the output layer. Thereafter I will refer to this model as the deep autoencoder. This model has several good properties in terms of its ability to train efficiently, reproduce the data

Table 3.2: Comparison of different models

Latent layer activation	Hidden layer activation	Epochs to train	Reconst-ruction error	Number of used components	Distributions of latent components
RELu	RELu	90	27.6932	64	Zero-inflated unimodal
ELU	RELu	70	27.7813	53	Unimodal
ELU	ELU	130	27.4960	54	Unimodal
ELU	Softsign	110	27.7570	50	Unimodal
ELU	Sigmoid	200	27.7050	58	Uni- and bimodal
ELU	Tanh	70	27.3830	98	Unimodal
Softsign	RELu	220	29.1858	8	Multimodal
Softsign	ELU	190	27.4026	93	Multimodal
Softsign	Softsign	520	28.0165	37	Uni- and bimodal
Softsign	Sigmoid	240	27.5861	78	Uni- and bimodal
Softsign	Tanh	110	27.6197	94	Uni- and bimodal
Sigmoid	Sigmoid	500	27.8981	40	Uni- and bimodal
Tanh	Sigmoid	350	27.6439	62	Uni- and bimodal

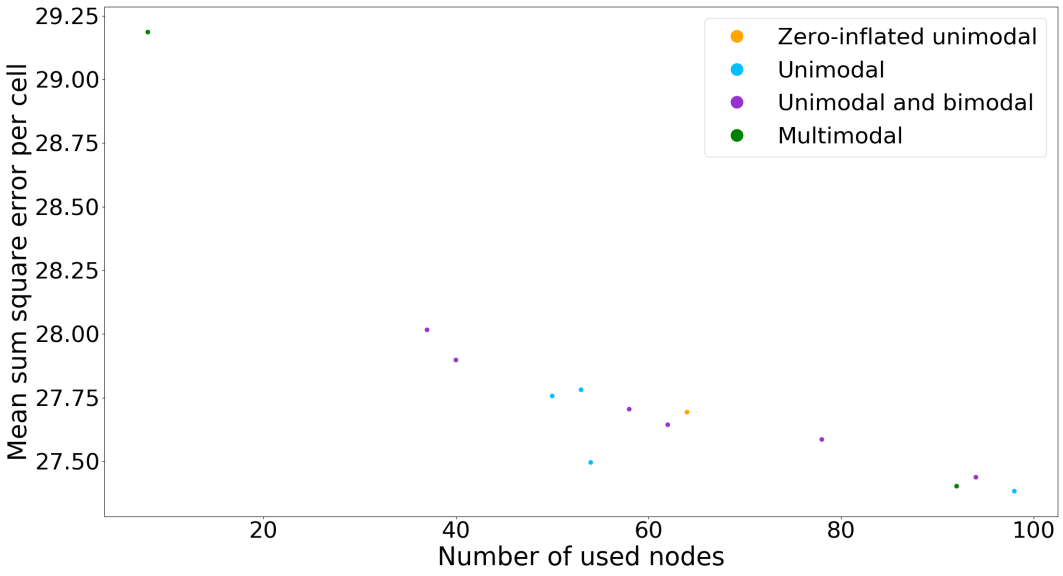
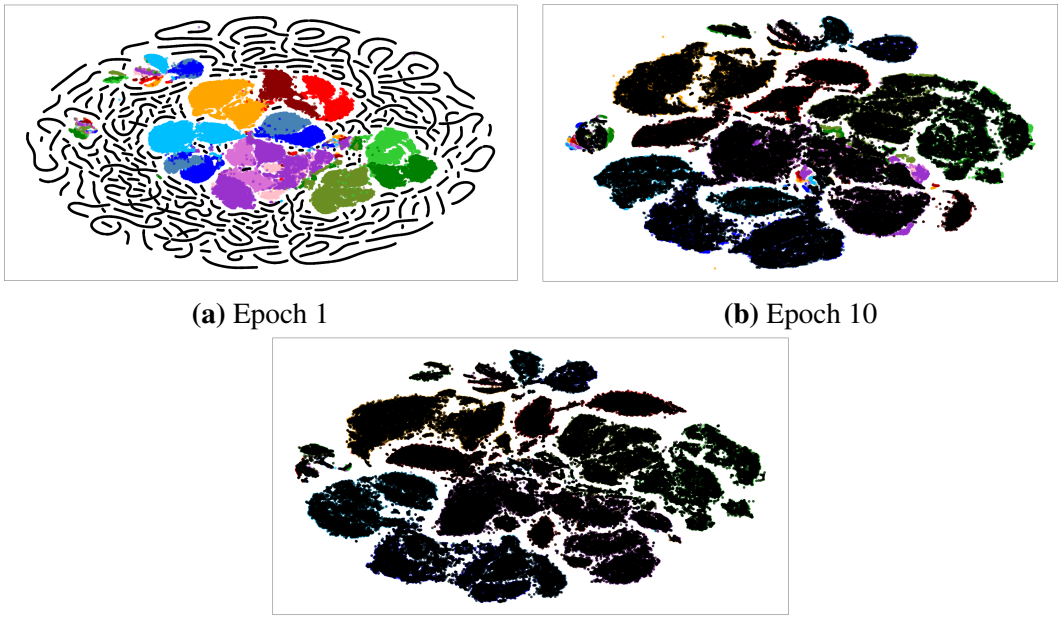


Figure 3.13: The relationship between the number of latent components used by a trained model and the reconstruction error. The plot is coloured by the the types of distributions of values in latent components.

well and create useful latent representation of the data. While it takes 190 epochs to train the model to optimality, it only takes 10 epochs to reduce the reconstruction error from 37.6578 to 29.4705, and further 10 epochs to reduce it to 28.6901. The following 170 epochs refine the model to achieve the reconstruction error equal to 27.4026, one of the lowest compared to all other models tested. The dynamics of the training can be visualised with a tSNE plot of both real and reconstructed data, see Figure 3.14. After the first epoch none of the reconstructed cells overlap with the clusters of real cells, after ten epochs there is a good overlap between the real and the reconstructed cells. The plots of the real and reconstructed data after 20 and 190 epochs are indistinguishable.

The deep autoencoder is an over-parametrised model, and hence if it is trained more than once on the same data the results will be different. To estimate the variability



(a) Epoch 1 **(b)** Epoch 10

(c) Epoch 20

Figure 3.14: tSNE plots of real and reconstructed data after 1, 10 and 20 epochs of training the deep autoencoder. The data is shown in colour (each sample in a different colour, similar to Figure 3.9), the reconstructed data is shown in black.

between training runs I trained the deep autoencoder on the Cheng et al. [2018] dataset 20 times. The reconstruction error produced across these 20 runs range from 27.4025 to 27.5628, with the mean equal to 27.4840 and standard deviation equal to 0.045. For comparison, reconstructing the data using 100 PCs produces an error equal to 27.6746, which is more than 4 standard deviation away from the mean of the reconstruction error of the autoencoder.

I've shown that the average reconstruction error is significantly lower for the deep autoencoder compared to 100 component PCA. It is important to know whether this difference is consistent (i.e. the reconstruction error is lower for most cells) or driven by a group of cells for which one of the methods works much better than the other. Figure 3.15 shows that the reconstruction error produced by the

deep autoencoder is lower than the one produced by the 100 PCs for 77.81% of the cells in the data. There is no group of cells for which the autoencoder works especially well or especially bad. The most interesting aspect of Figure 3.15 is that it shows the strong correlation between the reconstruction errors produced by the deep autoencoder and the 100 PCs. The Pearson correlation coefficient between the two is 0.9962, which suggests that there might be an underlying property of the expression profiles that makes some of them easier to reproduce than others. One apparent property is the number of genes expressed above 0.4 (in the data scaled to [0,1] range where 1 corresponds to the most highly expressed gene in a cell). The Pearson correlation coefficient between the number of genes expressed above 0.4 and the reconstruction errors produced by the deep autoencoder and the 100 PCs is 0.7587 and 0.7589 respectively. This implies that the more highly expressed genes there are in a cell, the "easier" it is to reconstruct the expression profile, regardless of which method is used. This makes sense intuitively - the expression values for genes expressed at low levels are dominated by noise, while the expression values for genes expressed at high levels are more informative (after log-transformation).

The latent space created by the deep autoencoder consists of 100 components (i.e. dimensions) bounded to (-1, 1) range, which allows for easy sampling from this latent space. Only 93 of these 100 components are used by the deep autoencoder, indicating that the capacity of the model is bigger than necessary and hence simply increasing the number of nodes in the latent layer will not improve the performance of the model. The variances of the components produced by the deep autoencoder,

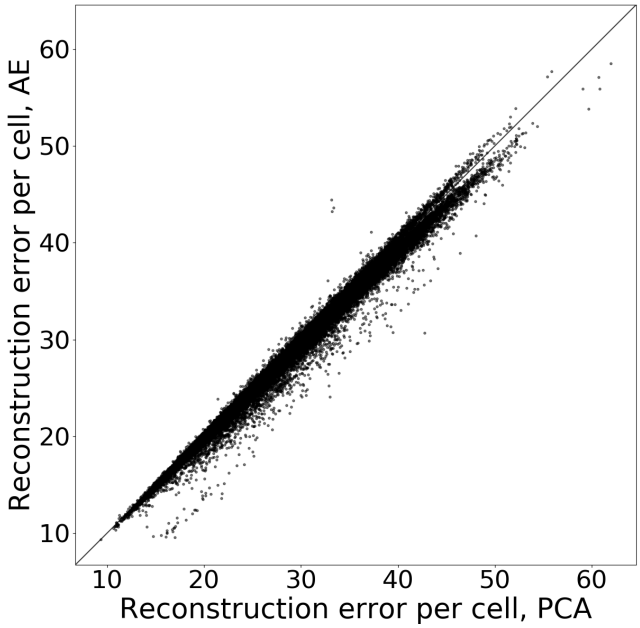


Figure 3.15: Comparison of reconstruction errors produced by the deep autoencoder and by the 100 PCs.

the basic deep autoencoder with ReLU activations in all but the output layer, and the PCA are not directly comparable as the components are bounded to different ranges. To accommodate for this, I first computed the variances of each component and then divided them by the maximum variance. Figure 3.16 shows that while amongst PCs there are only a small number of components with considerable variance, both autoencoders have much higher number of nodes with considerable variance (i.e. an informative embedding in that dimension). The deep autoencoder compares favourably with the earlier architecture that used ReLU activation function, as it has 93 components with an informative embedding of the data. Most importantly, the distributions of values in the 93 used components produced by the deep autoencoder are diverse - compare the distributions with one, two and three modes in Figure 3.27 to zero-inflated unimodal distributions produced by the basic deep autoencoder with ReLU activations in all but the output layer shown in Figure

3.12. To fully make sense of the latent representations of the data, I will first analyse the flow of the information through the deep autoencoder in Section 3.2.4 and then discuss the properties of the latent space in detail in Section 3.2.5.

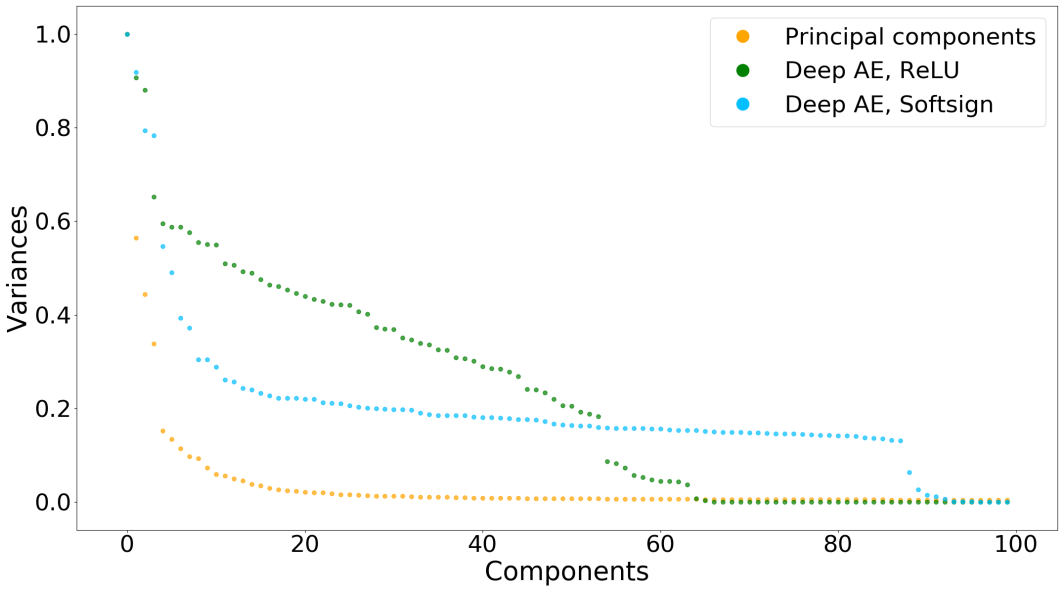


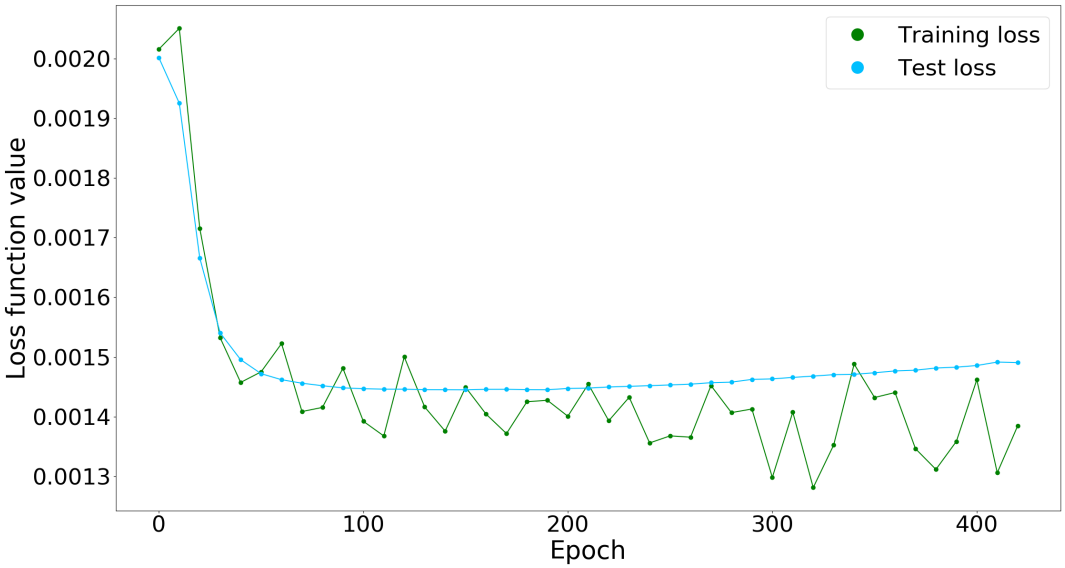
Figure 3.16: Comparison of variances of latent components produced by PCA and two autoencoders (AE) with different activation functions in the latent layer.

PCA identifies linear combinations of the gene expression values that correspond to the largest variance in the data. If none of the genes or linear combinations of the genes are able to explain the variance in the data, that would be immediately apparent. For example, the individual variances of the first five PCs of the Cheng et al. [2018] dataset are 2.3815, 1.3448, 1.0581, 0.8052 and 0.3612. If the gene expression values are randomly permuted for every cell, the individual variances of the first two PCs are 0.02521 and 0.00052. It is not immediately apparent with GNN-based models, since neural networks can be trained on noise [Arpit et al., 2017, Neto, 2018]. The plot of the loss function values for the deep autoencoder

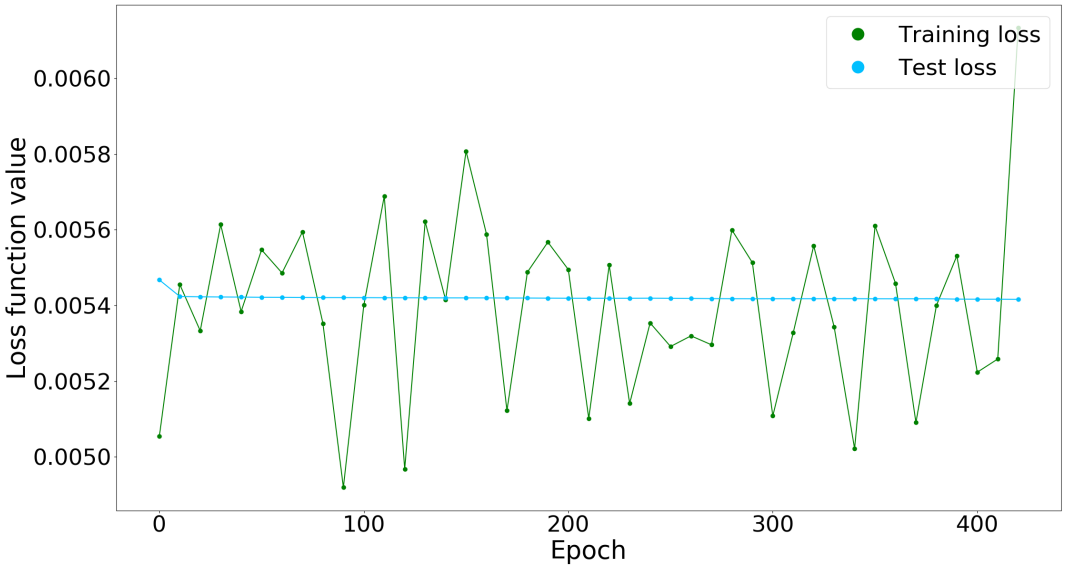
trained on the Cheng et al. [2018] dataset, and on the same dataset with the gene expression values randomly permuted for every cell, shows that the training dynamics are completely different, see Figure 3.17b. During the training of the deep autoencoder on the real data, the loss function for the batches of training data first decreases rapidly as every adjustment of an initially random mapping leads to an improvement, and then starts to oscillate as the model is further refined. For most of the training the loss function for the batches of training data is lower than the loss function of the test dataset, since the model has never observed the test dataset. The loss function for the test dataset decreases monotonically until the onset of overfitting and then increases monotonically afterwards. In contrast, during the training of the deep autoencoder on noise, the loss function for the batches of training data oscillates and its values are often higher than the initial value - the mapping after training is no better than an initially random mapping. The loss function for the test dataset remains largely constant since adjustments to the model are random instead of directed. These differences imply that the datasets with insufficient signal to noise ratio can be identified by monitoring the training dynamics.

3.2.4 Information flow through a GNN

In this section I will follow the flow of information through the deep autoencoder - starting with the input data, through the hidden and the latent layer of the encoder, the hidden layer of the decoder and finally the output layer of the decoder. Intuitively one would expect that hidden layers of a neural network extract informative



(a) Training on data



(b) Training on noise

Figure 3.17: Comparison of the loss function values for the deep autoencoder trained on (a) the real data and on (b) noise.

features from the input they receive. Huang et al. [2019] formalised this intuition by showing that for a particular application the features extracted by a hidden layer coincide with the result of a feature selection optimisation problem.

In the hidden layer of the encoder, a gene expression profile is first multiplied by a matrix of weights with the number of rows corresponding to the number of genes in the expression profile and the number of columns corresponding to the number of nodes in the layer, 500 in this case. Then, biases are added to each of the resulting 500 values. Finally, the resulting values go through the Exponential Linear Unit (ELU) activation function that is linear on the non-negative domain and non-linear on the negative domain. The weights and the biases are initiated with random values close to 0 and are adjusted through backpropagation. There are no restrictions on the values of the weights and the biases. In the deep autoencoder trained on Cheng et al. [2018] data the weights in the encoder hidden layer range from -0.6837 to 0.9734, 86.2% of the nodes have more negative weights than positive weights, see Figure 3.18. Figure 3.19 shows how for one example cell in the data the values of each of the 500 nodes is composed of the contribution of negative weights, positive weights and added biases. The values of biases range from -0.2705 to 0.1369; they are small relative to the values obtained by the nodes. The average number of encoder hidden layer nodes with negative values per cell for this specific dataset is 420 out of 500. This suggests that the network is exploiting the non-linear part of the domain. Finally, the ELU activation function applied to these node values transforms them to (-1, ∞) range. This results in the hidden representations of an

input cell - a 500 dimensional vector with values in the approximate range from -1 to 8. Examining the correlations between the weights associated with each of the 500 nodes reveals that the maximum absolute correlation between the weights associated with one node and another node is on average 0.1840. There is a striking correlation inflation as the information passes through the layer - the maximum absolute correlation between the node values corresponding to the hidden representations of the data is on average 0.7419. The values of 11 out of 500 nodes have an absolute correlation of more than 0.99 with values of another node. None of the nodes are immediately interpretable as corresponding to batch effect, cell cycle or other known characteristics of the cells. This makes sense intuitively - the training of the network was driven by a single optimisation problem to minimise the reconstruction loss value. There was no incentive for the network to partition different features of the data across different layers. Hence, the features only become apparent when the output of the whole encoder is considered, not just a single layer.

In their attempt to make sense of the features produced by a GNN-based model, TAN et al. [2014] focused their attention on the genes that are associated with weights that are more than two standard deviations away from the mean of the weights for each of the nodes. It's the magnitudes of the weights and not the distances from the mean of the weights that determine which genes have the most impact on the value of a node. To get an estimate of how many genes might have a major impact on the value of each of the nodes, I calculated the number of genes as-

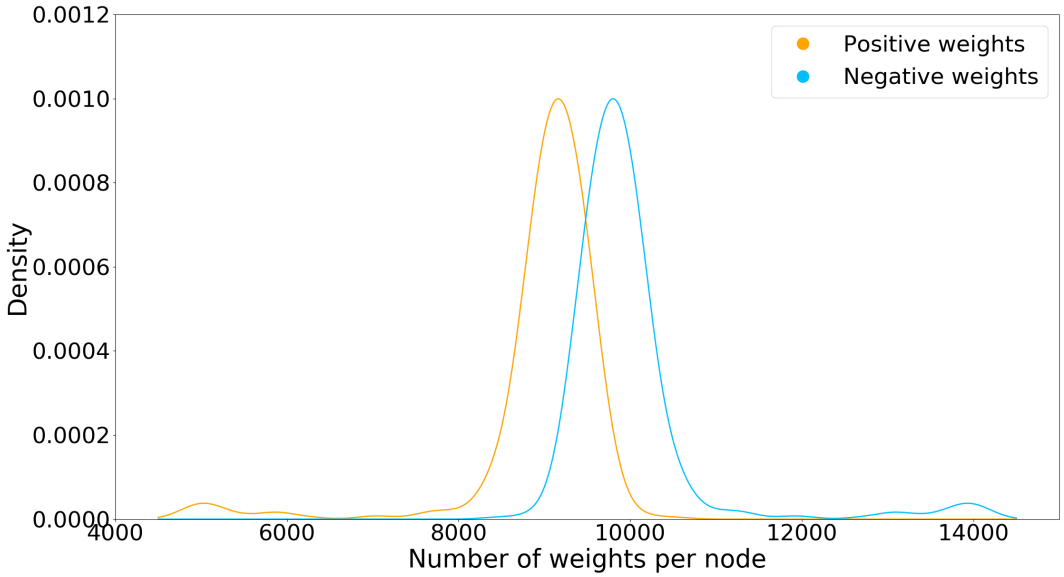


Figure 3.18: Positive and negative weights in the hidden layer of the encoder.

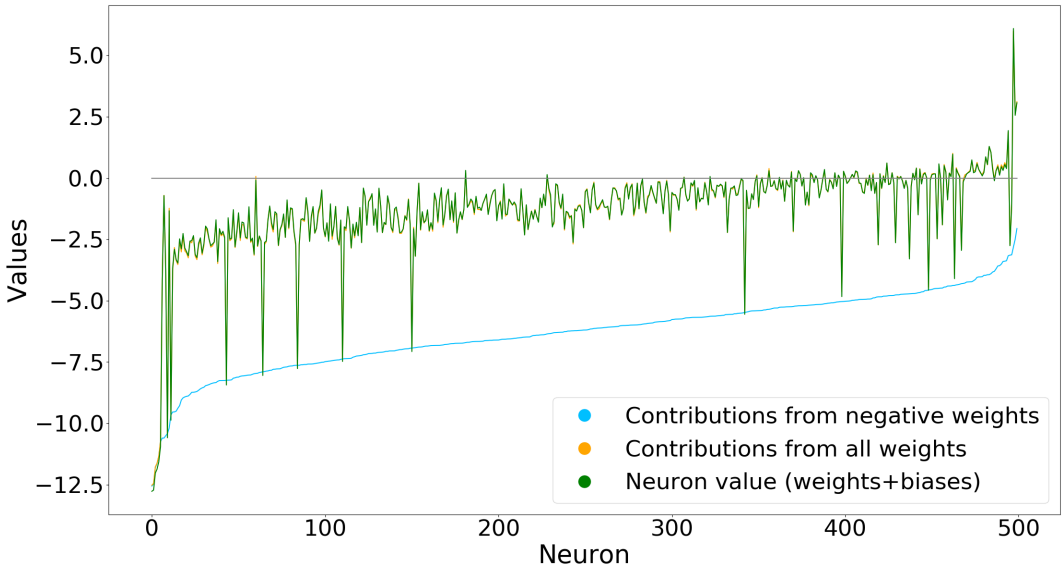


Figure 3.19: The values of each of the 500 nodes for one example cell in the data ordered by the contribution of negative weights. The values are composed of the contribution of negative weights, positive weights and added biases. The magnitudes of biases are small relative to the values obtained by nodes - the green line (the total values) mostly overlaps the orange line (the values without the biases).

sociated with the weights that are a certain number of standard deviation away from 0. Figure 3.20 shows that for an analysis concerned with a handful of known marker genes the cut-off value for the weights that are considered significant would have to be high - about 10 standard deviations away from the mean value of the weights. The distribution of weights for each node is heavy-tailed, with most of the values clustered around the mean of the distribution and a lot of outliers spread many standard deviations away from it. The impact of the genes associated with high positive weights is different from that of the genes associated with high negative weights. Due to the properties of the ELU activation function, if a gene is associated with a high positive weight and the resulting value of the node is positive, then the effect of this gene is linear and unbounded. In comparison, if a gene is associated with a high negative weight and the resulting value of the node is negative the effect of this gene is non-linear and bounded. Genes that are on average more highly expressed have more effect than genes with lower expression.

To estimate an impact of each of the variables (genes, in this case) on the model performance, Breiman [2001] introduced a variable importance method that calculates importance scores for each variable in the training data based on how much difference it makes if a variable is replaced with noise. The major flaw in this approach, as pointed out by Hooker and Mentch [2019], is that the importance scores mostly depend on the model's ability to extrapolate as replacing a variable with noise creates data points that do not occur in reality. For example, consider a gene that is either not expressed or highly upregulated. Replacing it with noise would

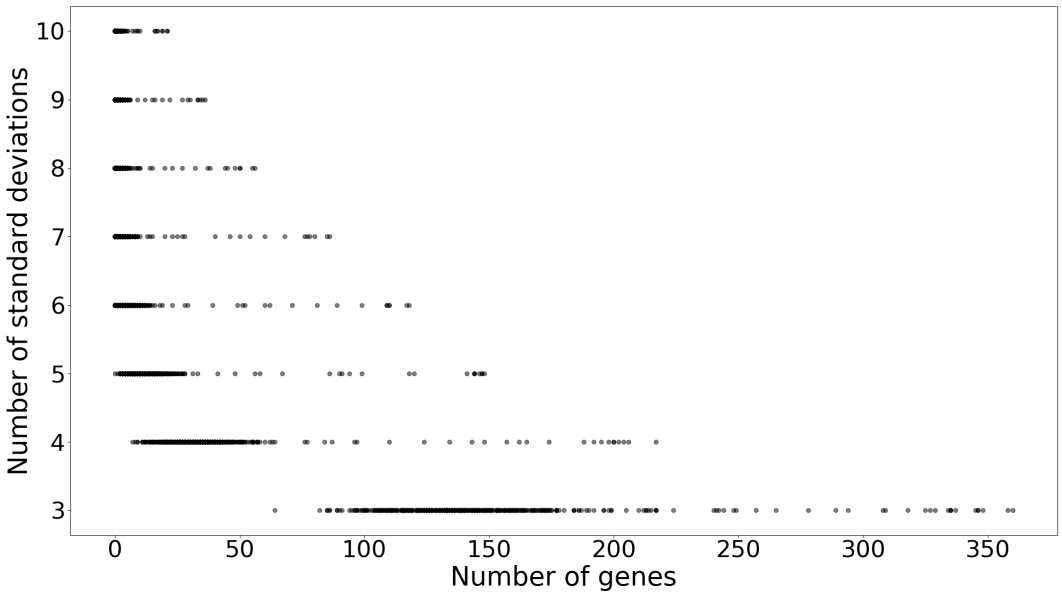


Figure 3.20: The number of genes associated with the weights that are a certain number of standard deviation away from 0 for each of the nodes.

create cells where this gene is upregulated in a biologically unrealistic context. Instead, I used another approach - for each gene I identified the cells in which it is expressed and compared the values of the nodes resulting from using the real data and the real data with the expression of this gene set to 0 as an input. I defined the gene's impact score as the mean of the absolute differences between the values for each cell. Based on the distribution of these impact scores, see Figure 3.21, I chose an arbitrary cut-off value 0.024 to identify genes that have a major impact on each of the nodes. Values of 321 nodes receive major impact from 1 to 10 genes, values of 113 nodes from 11 to 116 genes, and for 66 nodes no genes have major effect on their value, see Figure 3.22.

From the 18962 genes in the dataset only 418 genes have a major impact on at least one node, and 307 have a major impact on more than one node. There is

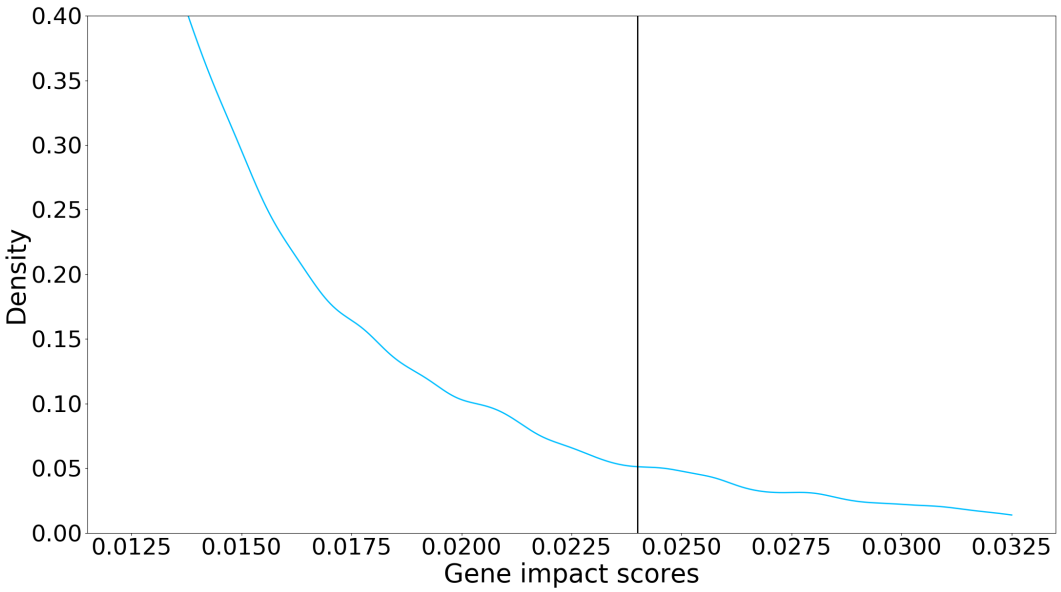


Figure 3.21: Distribution of the gene impact scores.

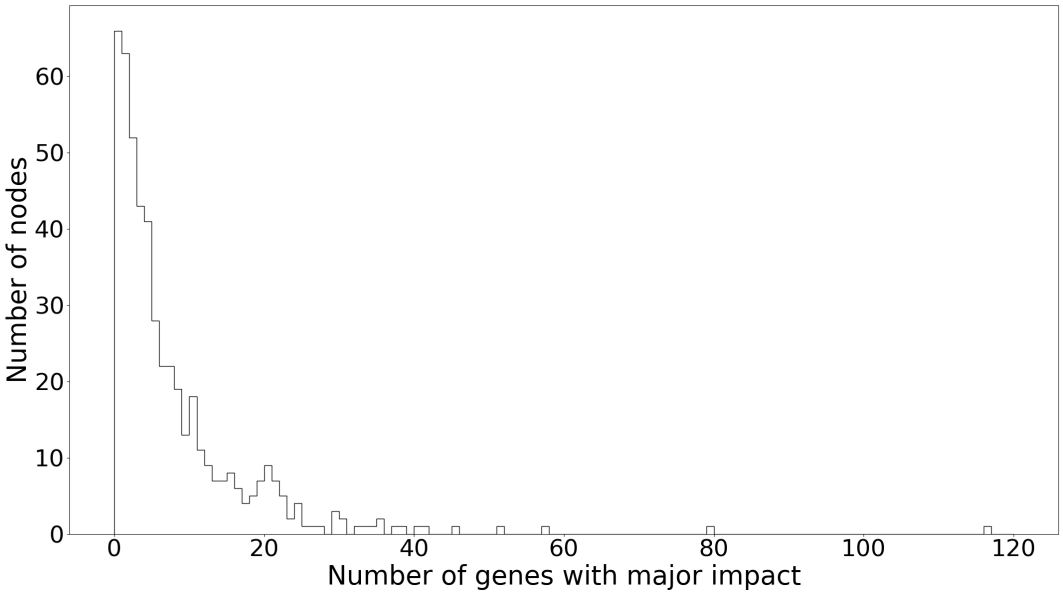


Figure 3.22: The number of genes with a major impact on the value of one or more nodes.

a relationship between the maximum expression value associated with a gene in the dataset and the number of nodes on which this gene has a major impact, see Figure 3.23. Only 13.64% of 3064 genes with a maximum expression above 0.4 have a major impact on at least one node. Using the average expression of a gene or a number of cells in which a gene is expressed as a predictor of the number of nodes it influences does not show a clear pattern. Intuitively one would expect that the number of nodes on which a gene has a major impact would be a predictor of the reconstruction error of expression values of this gene since more information about the expression of this gene is captured by the neural network, but this is not the case. To test the collective importance of the genes that don't seem to have a major impact on any of the nodes, I removed those 18544 genes, trained the deep autoencoder on the 418 remaining genes and compared the reconstruction error for those genes. The autoencoder trained on 418 genes only is able to reconstruct their expression much better than the autoencoder trained on the whole expression profiles - the reconstruction errors are 0.6487 and 1.4302 respectively.

The latent layer of the encoder is similar to the hidden layer - the input (in this case, 500 dimensional representations of the cells) is first multiplied by a matrix of weights, then the biases are added, and finally the activation function is applied. Softsign activation function used in this layer transforms the values of the 100 nodes to (-1,1) range. In the deep autoencoder trained on the Cheng et al. [2018] data the weights in the encoder latent layer range from -0.5405 to 0.6448. There is no bias towards negative weights, see Figure 3.24. The relationships between the 500 input

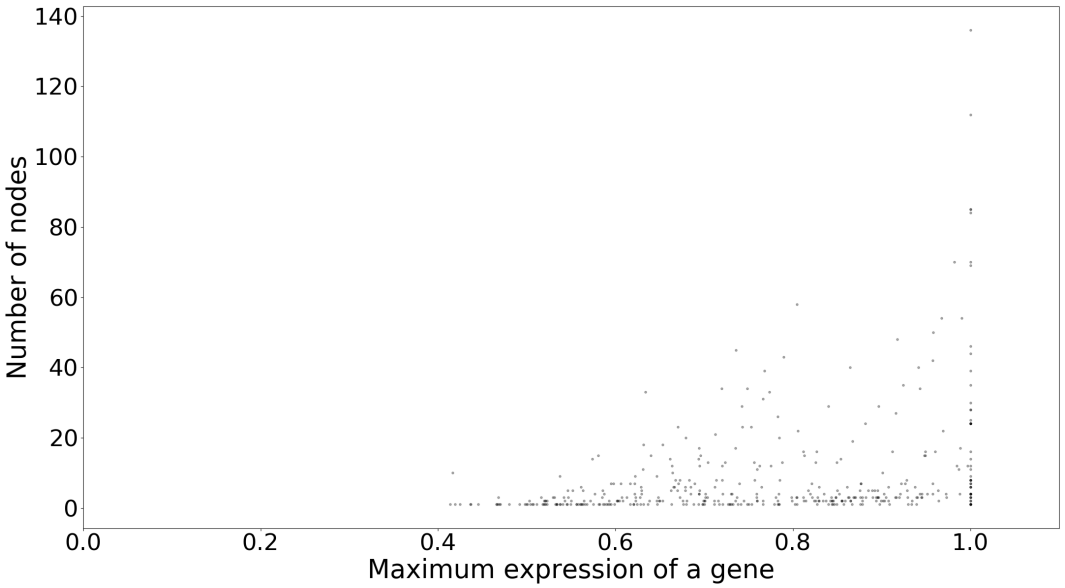


Figure 3.23: The relationship between the maximum expression value obtained by a gene in the dataset and the number of nodes on which this gene has a major impact. Only 418 genes that have mayor impact on at least one gene are shown.

features and the 100 nodes in the latent layer exhibits no easily interpretable pattern - the values of the nodes are not dominated by one or two features. The highest and the lowest weight in each of the 100 nodes in this layer are associated with one of the 140 input features, which implies that this layer utilises the information from the input effectively instead of focusing on a small number of most informative features. Unlike the hidden layer, in the latent layer the correlation inflation is not observed - the average maximum correlation between the weights of the nodes is 0.2086 and between the values it is 0.2934.

As shown in Section 3.2.3, some of the components of the latent representation are readily interpretable. For example, component 40 identifies suprabasal cells. The value of this component is most influenced by hidden layer nodes 157 and 250. In turn, there are 22 genes that have major influence on the value of hidden layer

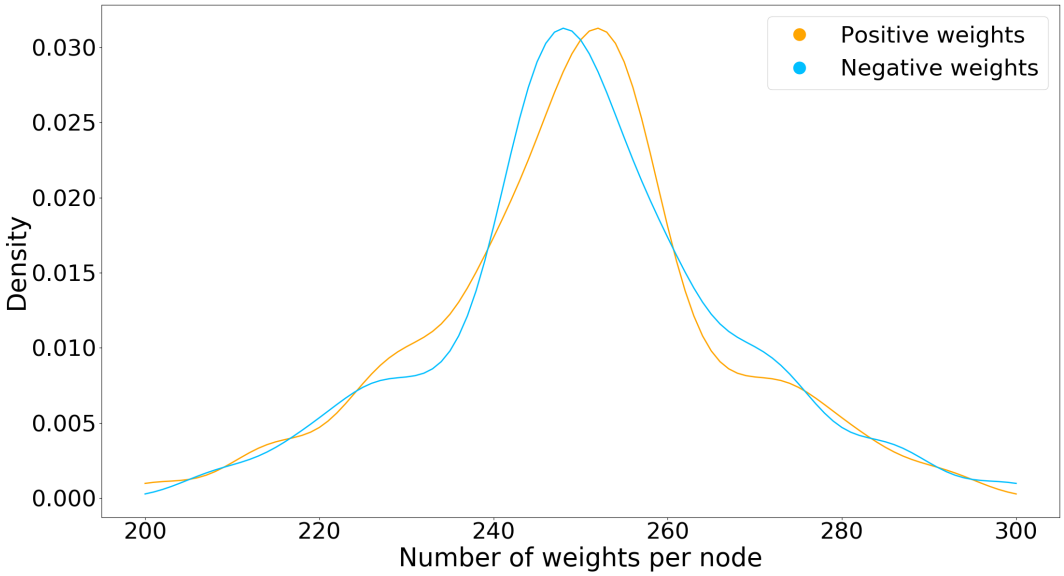


Figure 3.24: Positive and negative weights in the latent layer of the encoder.

node 250 including four genes from the keratin family KRT1, KRT10, KRT15, KRT14. There are 4 genes that have major influence on the value of hidden layer node 157 - KRT5 and KRT14 from the keratin family, suprabasin gene SBSN, and KRTDAP gene that might be related to the regulation of keratinocyte differentiation and maintenance of stratified epithelia. As expected, the values of components 51 and 38 that separate data by batch are influenced by a combination of many hidden layer nodes that are each in turn influenced by numerous genes.

The decoder hidden layer takes values of the 100 latent nodes as input and produces 500 features as output. The weights in this layer have a much wider range - from -2.0068 to 3.2268. The balance between positive and negative weights is similar to the encoder latent layer. The highest and the lowest weight in each of the 500 nodes in this layer are associated with one of the 87 latent nodes, excluding six unused nodes with the lowest variances. This implies that this layer utilises the information

from the input effectively and places little emphasis on unused latent nodes. ELU activation function used in this layer transforms the values of the nodes to (-1, ∞) range. The values obtained by the nodes in this layer lie in the approximate range from -1 to 9, which is similar to the range of values of the encoder hidden layer. In this layer the correlation inflation is not observed - the average maximum correlation between the weights of the nodes is 0.3986 and between the values it is 0.4614. The average maximum correlation between the values of the decoder hidden layer nodes and the encoder hidden layer nodes is 0.4434. It is not unexpected that in an overparametrised neural network the features identified by the hidden layers of the encoder and the decoder are different.

The decoder output layer is responsible for expanding the dimensionality back to the original number of dimensions, i.e. genes. The weights in this layer range from -1.1479 to 1.1018. As expected, 92.98% of the nodes have more negative weights associated with them, see Figure 3.25. The Sigmoid activation function transforms negative values of the nodes to gene expression values below 0.5; the majority of the data is in that range. The weights associated with each of the nodes provide no clues about the relationships between the genes. For example, the correlation between the expression values of the two keratin family genes KRT5 and KRT14 (the genes that have a major impact on the value of the encoder hidden layer node 250 that subsequently allows the encoder latent layer node 40 to identify suprabasal cells) is 0.8544. The correlation between the expression values of these two genes in the output of the decoder is even higher at 0.9158, but the correlation between

the weights of the decoder output layer nodes corresponding to these two genes is only 0.1317. The activation function used for this layer is non-linear but monotonic, which means that the relationship obscured by the non-linearity could be captured by Spearman rank-order correlation. In this case, the Spearman rank-order correlation is 0.1300, similar to the Pearson correlation. This is especially striking given that the average maximum correlation between the weights of the decoder hidden layer nodes is 0.6123.

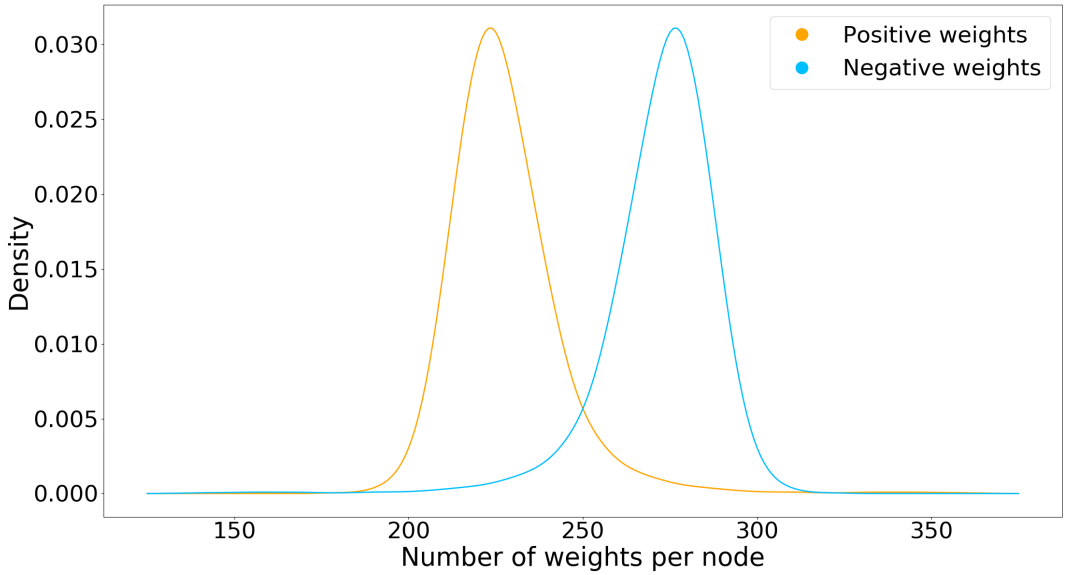


Figure 3.25: Positive and negative weights in the output layer of the decoder.

From this analysis of the flow of information through the deep autoencoder I conclude that the weights in the encoder can be used to interpret the latent dimensions created by the trained neural network, while the weights in the decoder cannot be used to infer the relationships between the genes. The main limitation of interpretability of the encoder weights is the distributions of the weights. These distributions have a narrow peak around the mean and often have no clear outliers.

The deep autoencoder training is governed by the loss function that is focused solely on the quality of the data reproduction and hence there is no incentive to create sparse weight matrices. As expected, the deep autoencoder exploits the non-linearity of the activation functions and places the most emphasis on the most variable features of the latent layer.

3.2.5 Data representation in latent space

The encoder of the deep autoencoder creates the embedding of the data into a lower dimensional space. A good embedding is useful in many ways - it allows one to discover important features in the data, it provides a lower dimensional coordinate space where similarity/distance between data points can be defined, and it facilitates the visualisation of the data by either picking two to three of the dimensions or defining combinations of these dimensions. An embedding of a single data point is produced by multiplying the expression profile by the encoder hidden layer weights matrix, adding the biases, applying an ELU activation function, multiplying the result by the encoder latent layer weights matrix, adding the biases and finally applying the Softsign function. Theoretically, each of the 100 dimensions of the latent space is interpretable - one can write an equation that shows the relationship between each of the gene expression values and the resulting value in the dimension of interest. In practice, for an expression profile with 10 genes such an equation would involve 55.6 thousand parameters and two non-linear functions. In this case, the expression profile contains 18962 genes which results in just over 9.5 million

parameters. For identifying important connections between the input genes and the resulting values in the latent vector it is necessary to use heuristics and/or existing knowledge, such as marker genes or known cell labels.

The latent space of the deep autoencoder consists of 100 dimensions in Euclidean space bounded to the (-1,1) range. A tSNE plot of the latent space is similar to the tSNE plot of the dataset, implying that the main signals in the data have been preserved in the latent space embedding, see Figure 3.26 and 3.9. The average maximum correlation between the values of the nodes is 0.2934. For comparison the average maximum correlation between 100 PCs is 0.0007. The embedding of the cells in the Cheng et al. [2018] dataset do not use the whole of the latent space. Instead they are spread over 55.43% of the range on average - the seven unused components use less then 7% of the range each, and other components use up to 77.56% of the range. The distributions of the unused components have sharp single-mode peaks around the boundaries of the domain. See Figure 3.27 for distributions of the latent components - 72 of them have a single-mode distribution centred around the middle of the domain, 13 have a bimodal distribution, and 6 of those 13 have the modes far apart. The other 8 components have various distributions - a trimodal distribution, single-mode distributions located close to the domain boundary, and distributions with unusual tail shapes. Components 95 and 10 that identify melanocytes and immune cells respectively have bimodal distributions with two peaks of unequal size located far apart, see Figure 3.28.

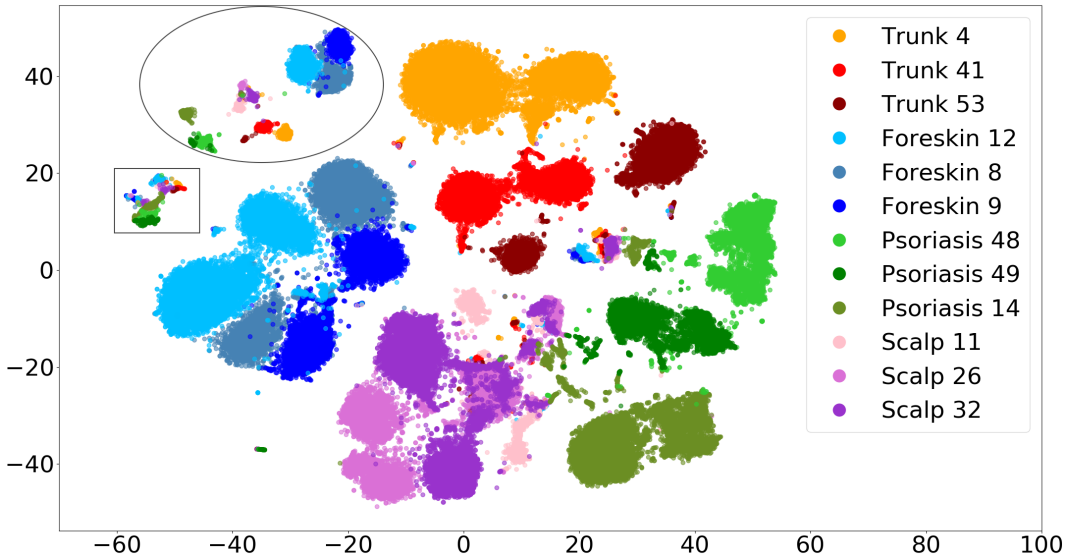
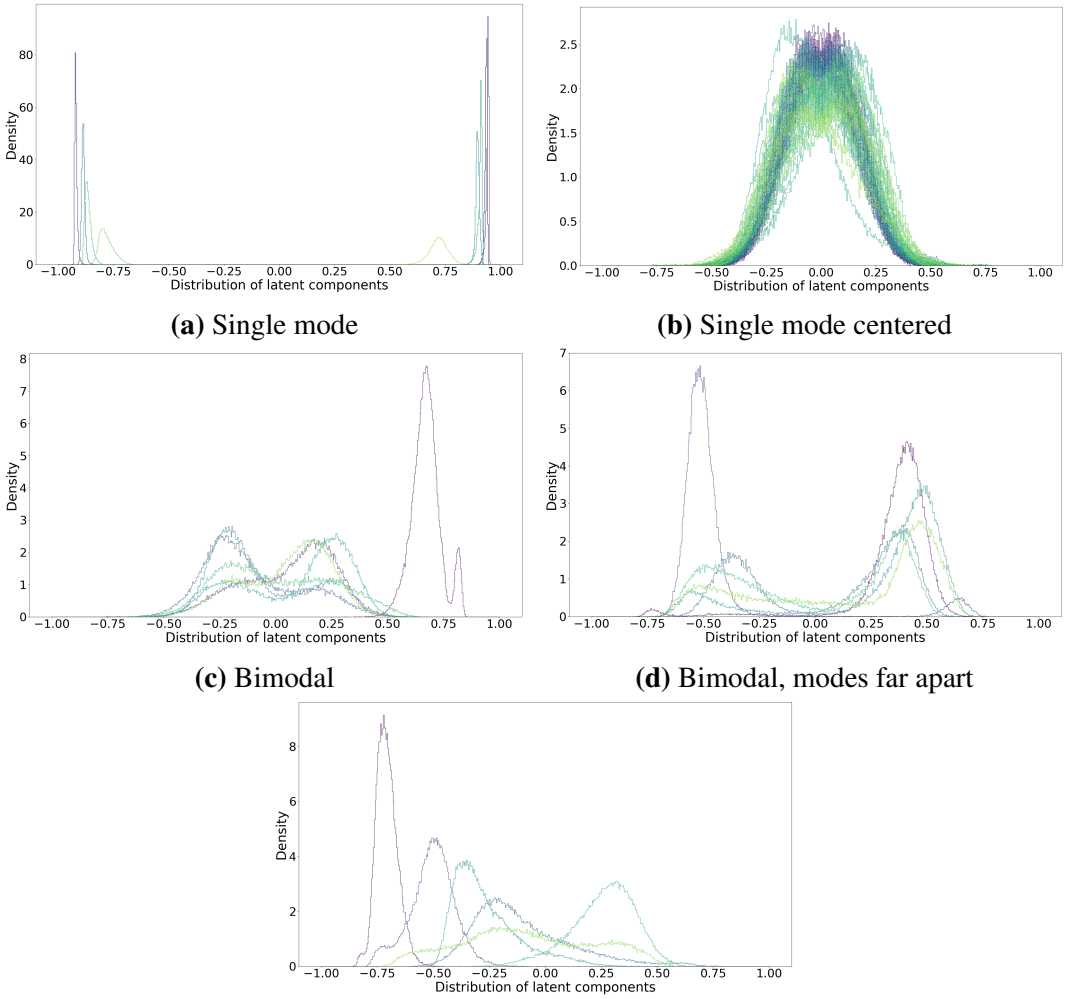


Figure 3.26: A tSNE plot of the latent space. Melanocytes (cells inside an oval) and immune cells (cells inside a rectangle) form separate clusters, while all other cells from every sample form 2 large clusters per sample. The cells are coloured by sample.

I identified latent components that correspond to cell cycle effects using the Pearson correlation coefficient between the values of a component and the total expression of M/G1 cell cycle phase marker genes identified by Macosko et al. [2015]. Five of the components (29, 76, 85, 92, 48) capture some of the cell cycle effects. Component 29 has a positively skewed distribution and the other 4 components have distributions located close to the domain boundaries, see Figure 3.29a. Four of the seven unused components (19, 31, 30, 52) also correlate with M/G1 cell cycle phase marker genes. A tSNE plot of these 9 components identifies small sub-populations within the data, and shows that the components associated with cell cycle phase are independent of batch effects, see Figures 3.29c - 3.29d.

Identifying latent components that correspond to batch effects requires a measure of association between a continuous and a categorical variable - the values of the



(a) Single mode

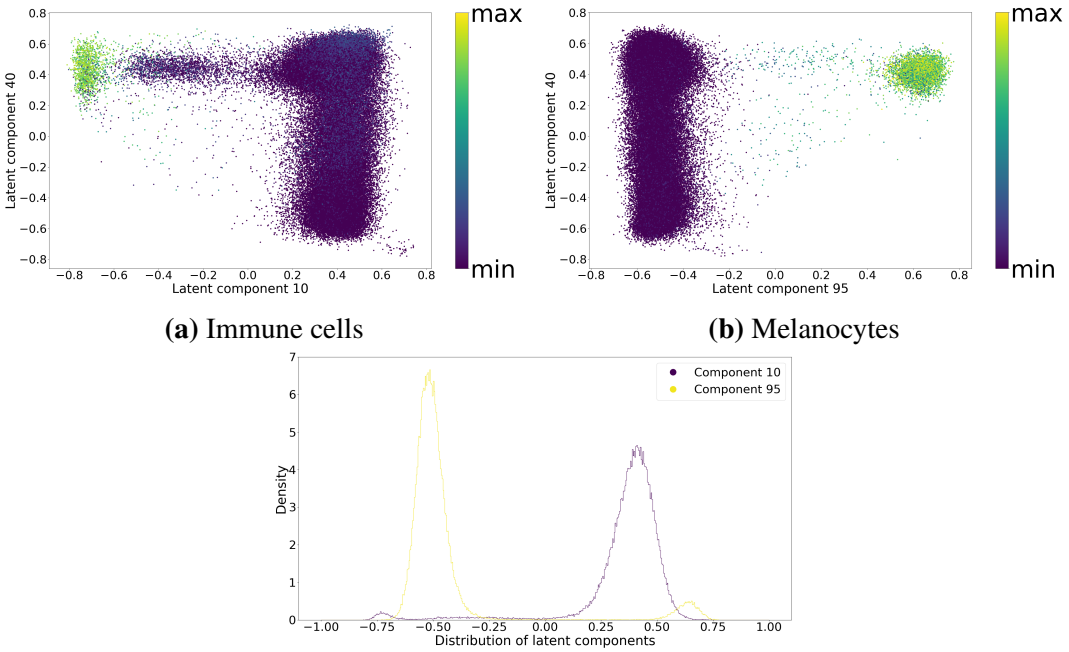
(b) Single mode centered

(c) Bimodal

(d) Bimodal, modes far apart

(e) Other

Figure 3.27: Distributions of the latent components: (a) 9 of them have distributions with sharp single-mode peaks around the boundaries of the domain, (b) 72 have a single-mode distribution centred around the middle of the domain, (c) 7 have a bimodal distribution, (d) 6 have a bimodal distribution with the modes far apart, (e) 8 have various distributions - a trimodal distribution, single-mode distributions located close to the domain boundary, distributions with unusual tail shapes. Each of the components is shown in a different colour simply for ease of visualisation.

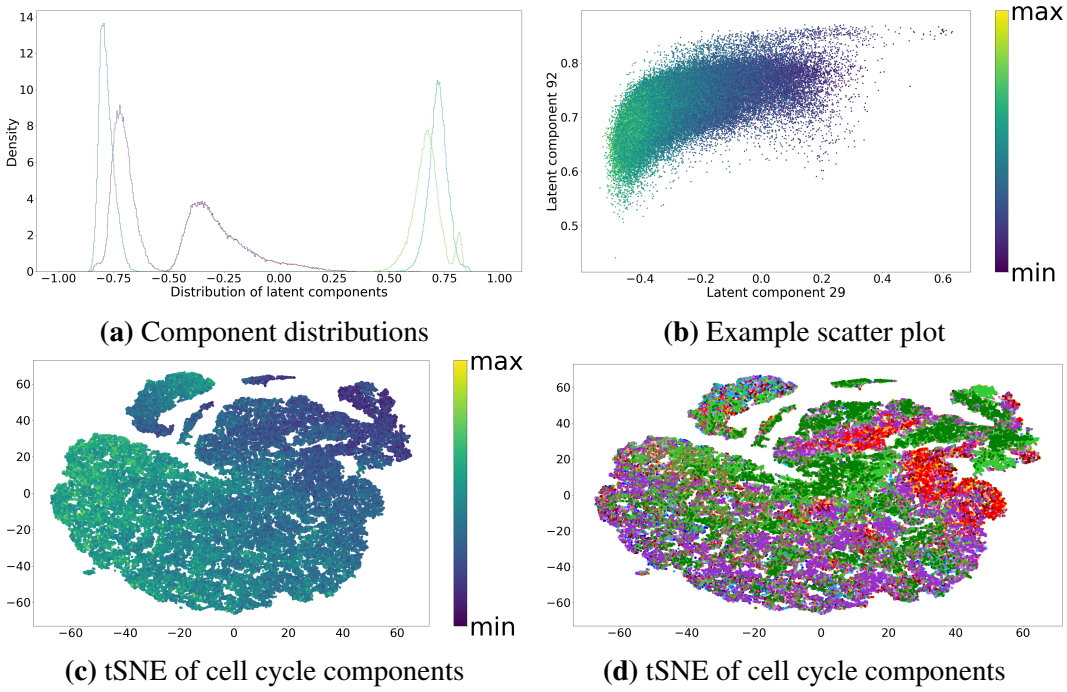


(a) Immune cells **(b)** Melanocytes

(c) Component distributions

Figure 3.28: (a) The components 10 identifies immune cells. The cells are coloured by the total expression of marker genes CD74 and HLA-DPA1. (b) The components 95 identifies melanocytes. The cells are coloured by the total expression of marker genes PMEL and TYRP1. (c) These components have bimodal distributions with two peak of unequal size located far apart.

component and the batch IDs. I used a simple approach based on an estimate of whether the variance of the continuous variable can be partially explained by the categorical variable. To do that, for each component I calculated the variance of its values associated with each of the batch IDs and the total variance of all the values. If the batch IDs are not related to the values of a component then the variances of the groups are expected to be similar to the total variance, which is the case for 92 components. If the variance of each individual group is lower than the total variance then the batch IDs explain some of the variance in this component. Eight of the components (51, 38, 17, 6, 22, 11, 21, 7) capture some of the batch effects. These components have distributions with one, two or three modes; all of



(a) Component distributions

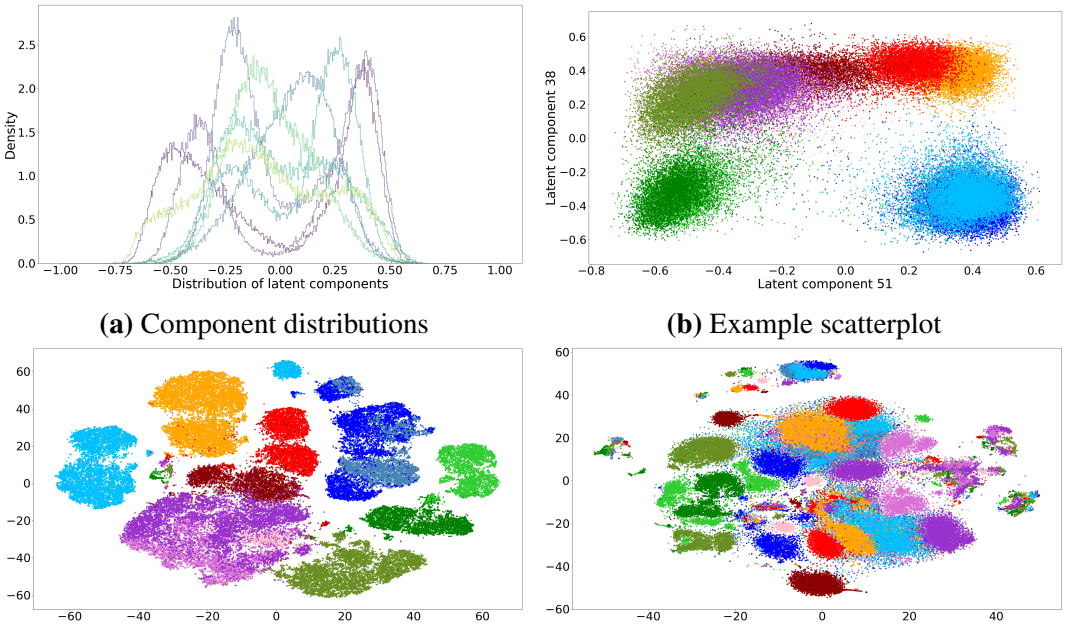
(b) Example scatter plot

(c) tSNE of cell cycle components

(d) tSNE of cell cycle components

Figure 3.29: (a) The distributions of the latent components 29, 76, 85, 92 and 48 that capture cell cycle effects. Each of the components is shown in a different colour simply for ease of visualisation. (b) An example scatter plot of two of these components. The cells are coloured by the total expression of M/G1 cell cycle phase marker genes. (c) tSNE of 9 components related to cell cycle phase. The cells are coloured by the total expression of M/G1 cell cycle phase marker genes. (d) tSNE of 9 components related to cell cycle phase. The cells are coloured by sample, similar to Figure 3.9.

them are located around the centre of the range, see Figure 3.30a. A tSNE plot of these 8 components shows that the batch effects are minimal between the three scalp samples and between the two foreskin samples, while the other samples are readily separated in the plot, see Figure 3.30c. There is also a sub-population of cells from all samples that forms a small cluster, implying that these cells might react differently to the stimuli that all cells are subjected to during the experimental protocol. A tSNE plot of 83 latent components that are not associated with batch or cell cycle phase effects shows that some batch effects are still collectively present amongst the remaining components, see Figure 3.30d.



(a) Component distributions **(b)** Example scatterplot

(c) tSNE of batch components **(d)** tSNE without batch and cell cycle effects

Figure 3.30: (a) The distributions of the latent components 51, 38, 17, 6, 22, 11, 21 and 7 that capture batch effects. Each of the components is shown in a different colour simply for ease of visualisation. (b) An example scatter plot of two of these components. (c) A tSNE plot of eight latent components associated with batch effects. (d) A tSNE plot of 83 latent components not associated with batch or cell cycle phase effects. In (b-d) the cells are coloured by sample, similar to Figure 3.9.

Before considering interpolation or extrapolation in the latent space or a subset of its components, I first assessed whether noise is distinguishable from data in this space. For this, I randomly selected 50 cells from the dataset and randomly permuted the gene expression values in each of the expression profiles. I will call them noise cells. These noise cells map to a single tight cluster in the latent space, see Figure 3.31. The position of these cells in individual latent components varies. They are located in the tail of the distribution of component 29 that corresponds to cell cycle, between the peaks corresponding to melanocytes and other cells in the distribution of component 95, outside the range of the unused component 55, and

throughout the range of the distribution of unidentified component 56, see Figure 3.32. Noise cells are not confined to the boundaries of the latent space and do not always fall outside the distribution of a latent component. There is no straightforward way to distinguish between an embedding of a cell used for training the deep autoencoder and an embedding of a noise cell.

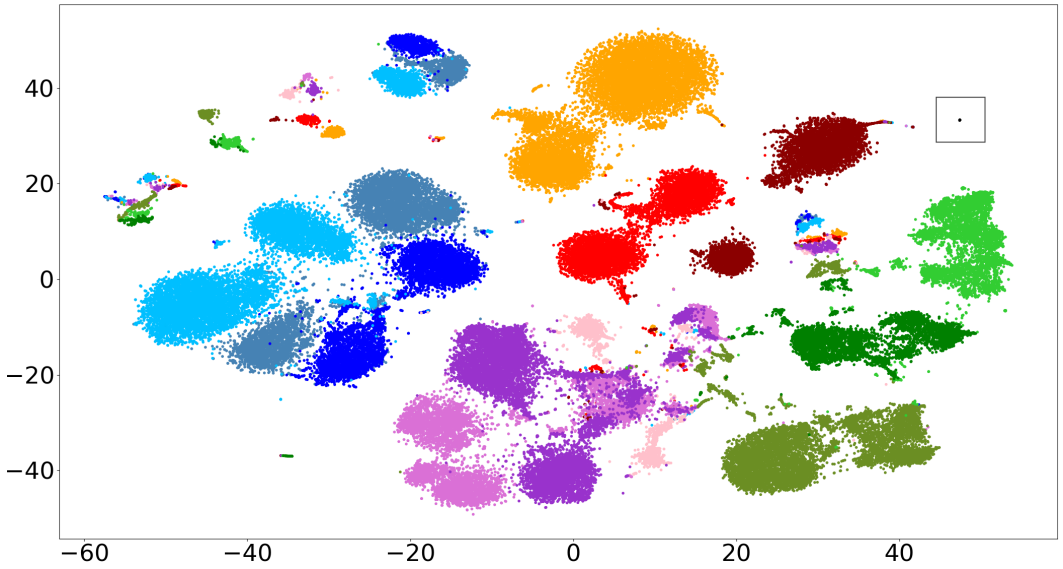
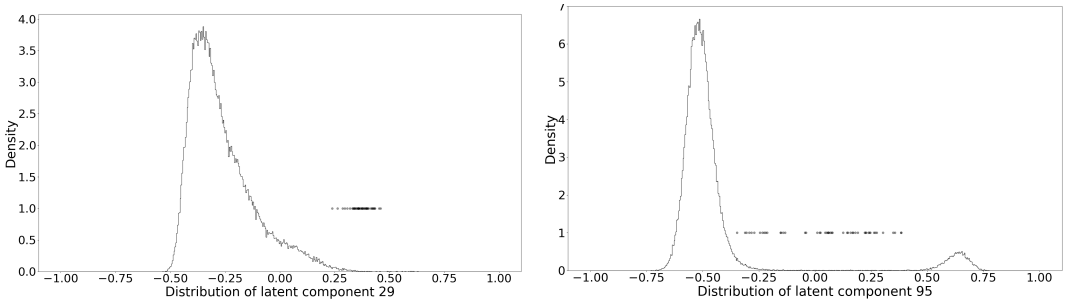
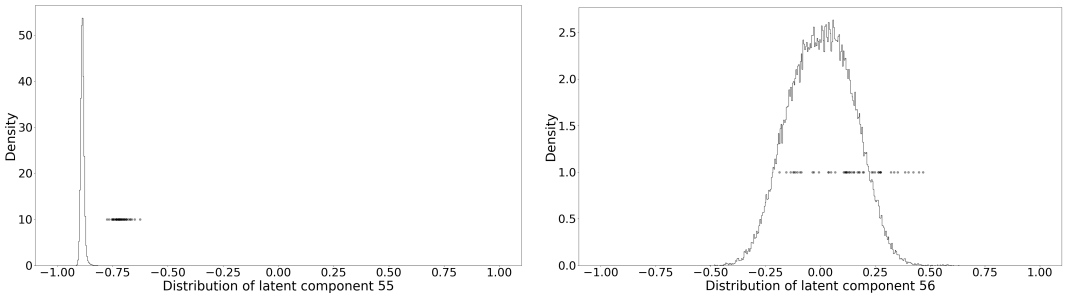


Figure 3.31: A tSNE plot of the latent space including both data and the 50 cells from the dataset with randomly permuted gene expression values in each of the expression profiles (enclosed in the rectangle). The data are coloured by sample, similar to Figure 3.9. The "noise" cells are in black.

Cheng et al. [2018] used 10 PCs of the data as input for Slingshot [Street et al., 2018] to infer differentiation trajectory from cells they've labelled as "basal1" to the ones they've labelled as "granular". Components 40 and 54 in the latent space created by the deep autoencoder correspond to the trajectory the authors of the data have reported, see Figure 3.33a. These components correctly identify immune cells as not part of the trajectory, but they are unable to separate melanocytes, WNT1, channel and follicular cells from the trajectory, see Figure 3.33b. Dong



(a) Cell cycle component **(b)** Melanocyte component

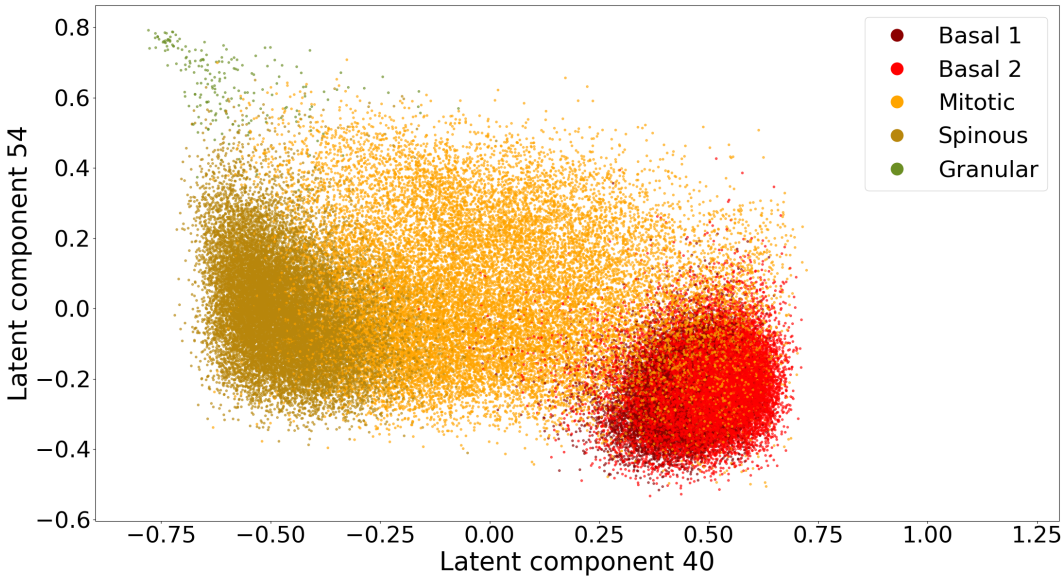


(c) Unused component **(d)** Unidentified component

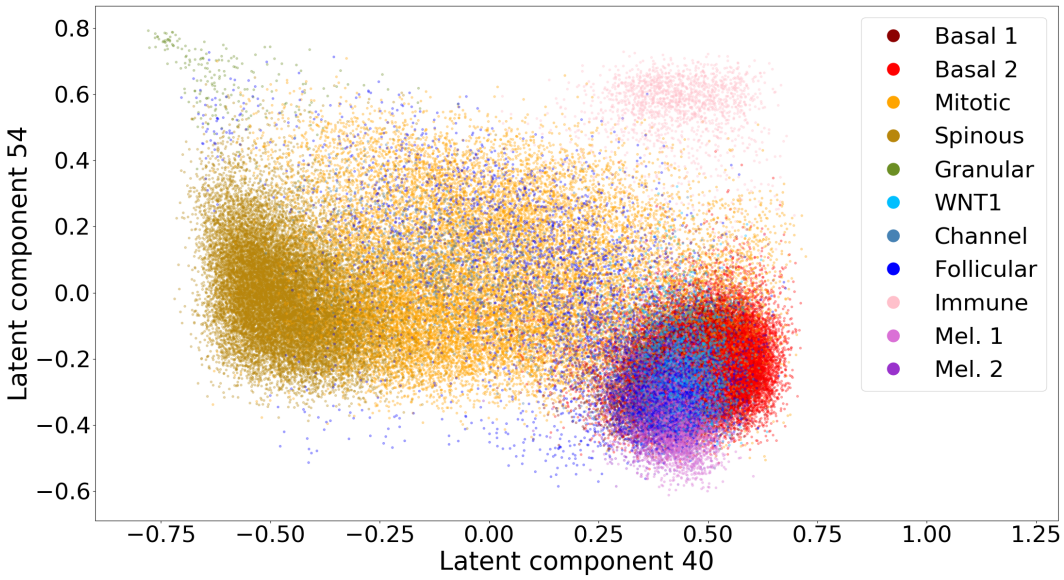
Figure 3.32: The distributions of the latent components 29, 95, 55 and 56. The values of these components for the "noise" cells (cells from the dataset with randomly permuted gene expression values) are shown with black dots.

et al. [2019] showed the inconsistency between feature representations in neural networks trained on images and semantic concepts. Similarly, it is likely that a neural network trained on gene expression profiles might produce features that are not aligned with the concepts that a biologist has in mind.

In light of these observations, I conclude that interpolation in latent space has to be considered with caution. Some of the latent component correspond to the trajectories in the data, for example components 40 and 54 capture the differentiation trajectory from basal cells to granular cells. Other latent components correspond to clusters, for example component 95 clusters the data into two groups - melanocytes and not melanocytes. While interpolation in the dimensions associated with the



(a) Cells on the trajectory



(b) All cells

Figure 3.33: (a) Components 40 and 54 correspond to the differentiation trajectory Cheng et al. [2018] have reported. Only cells that are part of this trajectory are shown. The cells are coloured by the cell type. (b) These components identify immune cells as not part of the trajectory, but they are unable to separate melanocytes, WNT1, channel and follicular cells from the trajectory. All cells are shown. The cells are coloured by the cell type.

trajectories in the data could be meaningful, interpolation in the "clustering" dimensions or in the dimensions corresponding to batch effects is not likely to be useful. Interpolation in the latent space is often used with the aim to infer propertied of the rare cell types or transient cell states, which means that the area of the latent space that is of interest is populated sparsely or not at al. While successful applications of vector arithmetics in latent space [Radford et al., 2015] are encouraging, the exact same examples also reveal the associated challenges - White [2016] showed that subtracting the smile vector from a woman's face resulted in an unintended addition of male attributes to the face. This problem originates from a sampling bias - the training data contained unequal proportions of images of smiling man and woman. This is a strong argument against using interpolation in the setting of unequal density of the data in the latent space. Most interpolation methods are based on the assumption that a differentiation trajectory, progression through a cell cycle or an infection, or any other path of interest corresponds to a straight line in the latent space. This assumption is rarely motivated by any evidence, and it is usually not possible to confirm the validity of this assumption in a straightforward way. Struski et al. [2019] argued that a good interpolation should follow the true distribution of the data. They proposed a method for constructing interpolations which takes into account the density of the data in the latent space and the differences between the reconstructions of consecutive interpolated points. They showed that this much more involved process performs better that the linear interpolation. This approach is based on the assumption that there is a continuous smooth path of data points along the whole interpolation trajectory. It would be

unjustifiable to make this assumption for the scRNA-seq data. A trained model cannot be expected to be predictive beyond the domain of the training data, which precludes the use of extrapolation in latent space.

3.2.6 A model useful beyond one dataset

A real test for the usefulness of a model is it's ability to generalise beyond one dataset it was trained on. To assess this, I used the human skin dataset produced by Tabib et al. [2018]. This dataset contains samples from dorsal forearm skin biopsies sequenced using the Chromium Single Cell protocol from 10X Genomics. The Tabib et al. [2018] dataset contains expression values for 17796 genes present in the Cheng et al. [2018] dataset, hence only 1166 genes had to be removed from the Cheng et al. [2018] dataset to be able to subsequently assess a trained model on both datasets.

The deep autoencoder trained on the Cheng et al. [2018] dataset using 17796 genes present in both datasets produced the reconstruction error equal to 26.9297 when assessed on Cheng et al. [2018] dataset. When assess on Tabib et al. [2018] dataset the reconstruction error is much higher - 35.8639. Similarly, the autoencoder trained on the Tabib et al. [2018] dataset produced the reconstruction error equal to 27.7581 when assessed on the dataset it was trained on and 34.5766 when assessed on the other dataset. The reconstruction error using 100 PCs computed on the Cheng et al. [2018] dataset is 27.1094 when assessed on the dataset itself and 35.0012 when

assessed on the other dataset. This implies that the features generated by the deep autoencoder generalise to other datasets no better than PCs.

3.2.7 Conclusions

In this chapter I've shown that the deep autoencoder has both theoretical and practical advantages over PCA and other dimensionality reduction algorithms. The deep autoencoder is able to capture non-linear interactions between the genes, and it does not assume that the regulatory relationships between the genes stay constant across the whole dataset. As a result, the reconstruction error is significantly lower compared to PCA. Though in terms of being useful beyond a single dataset on which it was trained, the deep autoencoder performs no better than PCA. The deep autoencoder also produces an embedding of the data into a latent space where more of the dimensions are informative. Across those dimensions, the deep autoencoder is able to generate a wide range of diverse distribution shapes that accommodate different types of features in the data. For example, a distribution with two distinct peaks located far apart distinguishes between immune cells and other cells, while a single-mode skewed distribution accounts for a continuous feature that has a "typical" state and a large range over which it can deviate. Another advantage of the deep autoencoder is that it does not train on noise, and hence if the data (due to a problem at experimental stage) contains no signal it is easy to determine it by looking at the training dynamics of the network. Unfortunately, the deep autoencoder does not allow meaningful interpolation or extrapolation between data points.

Chapter 4

Extending applications of GNNs to single cell RNA-seq data

In this chapter I will look at each of the components of the process of applying generative neural networks (GNNs) to the scRNA-seq data and explore the opportunities to improve upon each of them. The components of this process include the data itself, the internal architecture of the autoencoder, the dynamics of the autoencoder training and the inherent randomness property of the neural networks. Hereafter I will refer to the deep autoencoder model I explored in the previous chapter as the original autoencoder to distinguish it from the alternative implementations proposed below. The contributions made in this chapter are as follows:

- A novel approach to building an autoencoder robust to technical dropouts in the data
- A novel approach to model biological information flow in the cells using residual connection

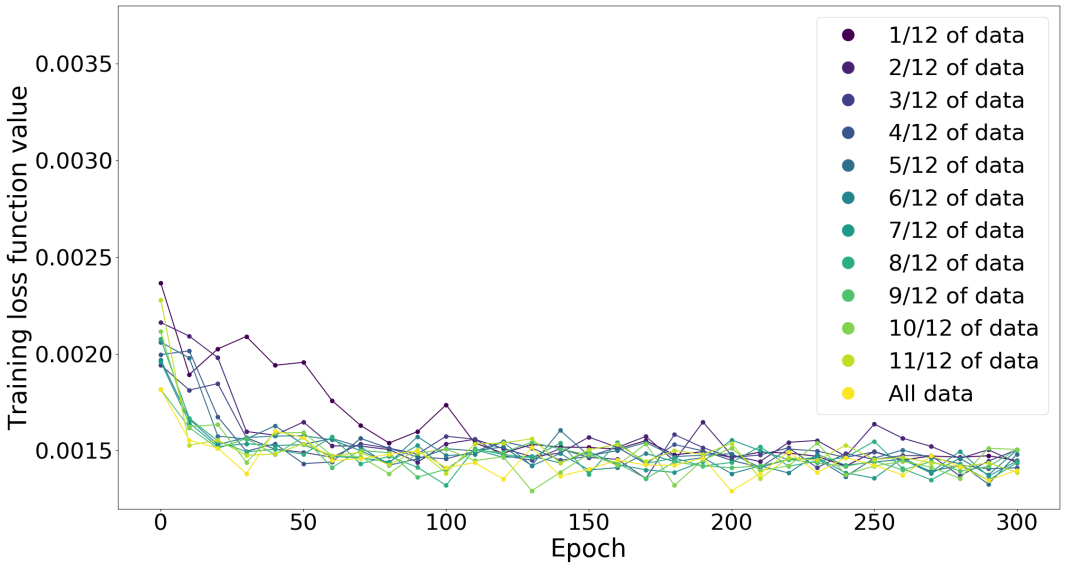
- An exploration of various approaches aimed at shaping the training process of the autoencoder

4.1 Addressing properties of single cell RNA-seq data

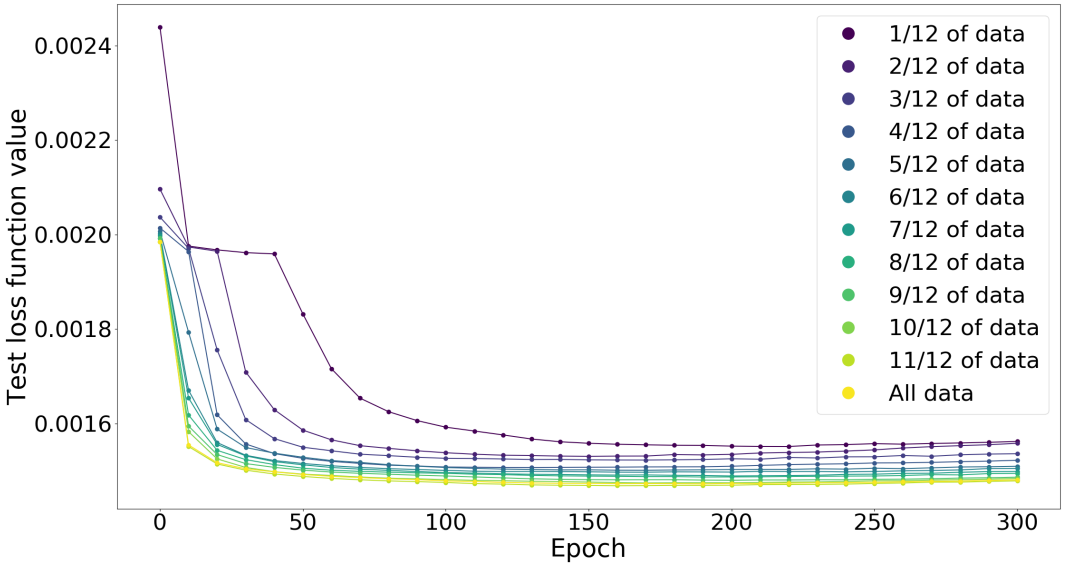
In this section I will explore the opportunities of improving the performance and the biological relevance of the autoencoder solely by manipulating the input data. First, I will assess the effect of increasing the amount of training data. Next, I will look for an appropriate transformation of the data to enable an autoencoder to learn from it more efficiently. I will conclude with a preposition of a novel method that could enable training a model that is robust to the technical limitations confounding the scRNA-seq data.

4.1.1 More data is always better

Often a simple way to improve the performance of a method is to use more data. To test whether the deep autoencoder would perform better if trained on more data, I randomly split the Cheng et al. [2018] dataset into 12 equal subsets and trained the model using 1 or more of these subsets. The dynamics of the training observed through the loss functions is similar across the 12 training rounds. The value of the training loss function after the first epoch is generally lower if more data is used for training, see Figure 4.1a, which is expected - more data means more learning during each epoch. After the first 30 epochs the training rounds are indistinguishable, apart from the model trained on only 1 subset of the data which takes longer to train. The



(a) Training loss function values



(b) Test loss function values

Figure 4.1: Comparison of the (a) training and (b) test loss function values. The colours show the proportion of the data used for training.

values of the test loss function are lower for the models trained on more data, but the difference is diminishing as more data is added, see Figure 4.1b.

For each of the training rounds I measured the reconstruction error on the subset of the data the model was trained on, as well as on the whole dataset, see Figure

4.2. The reconstruction error on the subset of the data used for model training reduces as more data is used for training, but the added benefit of each additional lot of data is diminishing. The reconstruction error on the whole dataset reduces as bigger proportion of this data was used for model training. The observed trend implies that if the experiment would have profiled more cells the improvement in the performance of the model would likely be modest. The additional information contained in the cells from the same experiment is limited.

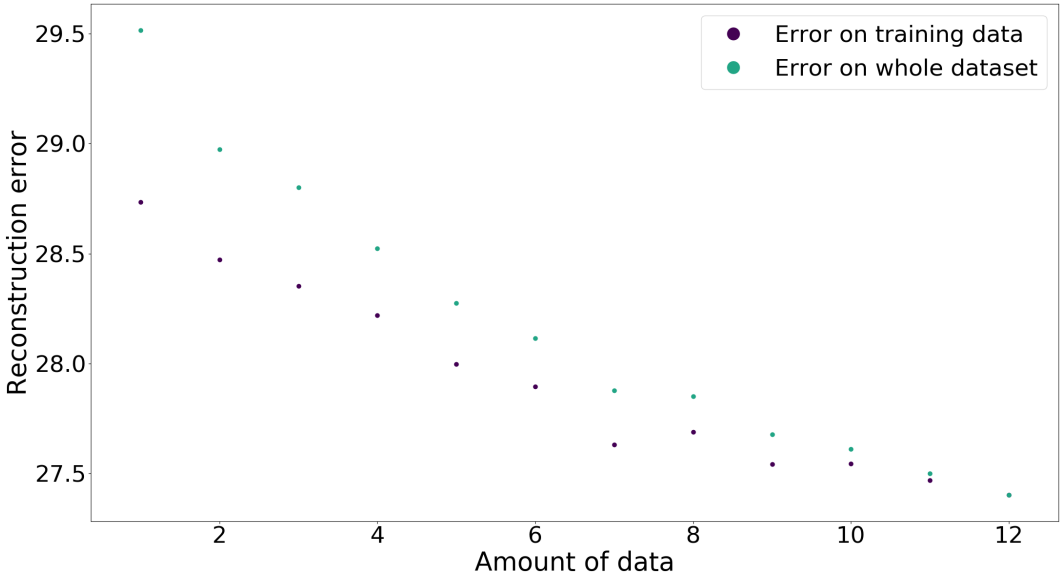


Figure 4.2: The reconstruction error measured on the whole data and only the data used for training the model. The x-axis show how many of the 12 subsets of data were used for training.

Another important question is whether the deep autoencoder is useful for combining datasets from different experiments produced by different labs. As shown in Section 3.2.6, the features identified by the deep autoencoder trained on one dataset generalise poorly to another dataset. This does not mean however, that the deep autoencoder is not useful for integrating information from several datasets. The

deep autoencoder trained on the Cheng et al. [2018] dataset (using only the genes present in both datasets) produces the reconstruction error equal to 26.9297. If both Cheng et al. [2018] and Tabib et al. [2018] datasets are used for training, the reconstruction error on the Cheng et al. [2018] data reduces to 26.8911. The effect on the smaller dataset is much more pronounced. The Tabib et al. [2018] datasets contains only 6497 cells, less then 10% of the number of cells in the Cheng et al. [2018] dataset. The deep autoencoder trained on the Tabib et al. [2018] dataset produces the reconstruction error equal to 27.7581. If both datasets are used for training, the reconstruction error on the Tabib et al. [2018] datasets reduces to 25.9300. This shows that the autoencoder is able to learn from a dataset produced by a different lab and integrate the information to improve the performance on the data of interest.

4.1.2 Making the data easier to learn from

While getting the desired amount of data is not always possible, it is always an option to scale or transform the data to enable GNNs to learn from it better. In this section I will explore whether the performance of the deep autoencoder can be improved by changing the range of the data in combination with selecting appropriate activation functions.

The deep autoencoder trains by minimising the loss function equal to the mean square error between the input and the reconstructed data. Since both the data and the reconstructed data lies in the [0,1] range, the square error of each individual

reconstructed gene expression value also lies in the [0,1] range. The derivative of the x^2 function is $2x$ and hence, the [0,1] range corresponds to the most shallow gradient of the loss function. To take an advantage of this property of the loss function used, the data could be simply scaled to a wider range. This would result in a larger maximum possible error and hence would provide bigger gradients for big errors and relatively smaller gradients for small errors, thus enabling the model to learn more efficiently, at least in theory. To test whether this can provide a performance improvement in practise, I scaled the data to [0,6] range and used the ReLU6 activation function in the output layer:

$$\text{ReLU6}(x) = \min(\max(0,x),6).$$

The choise of a [0,6] range is of course arbitrary, but the advantage of this specific choice is that there is no need to write a custom activation function since ReLU6 is included in the PyTorch library. Apart from the benefit described above, using a ReLU-based activation function in the decoder output layer will result in zero-inflated distribution of values in the reconstructed data which is ideal for scRNA-seq data - in the Cheng et al. [2018] dataset, for example, 87.64% of the gene expression values are equal to 0. To be able to compare the performance of this model, the reconstruction errors were computed on the data in the original [0,1] range. Contrary to the theoretical prediction, this modification does not result in an improved performance of the deep autoencoder - the reconstruction error increases significantly from 27.4840 to 27.7588 (more than 6 standard deviations from the

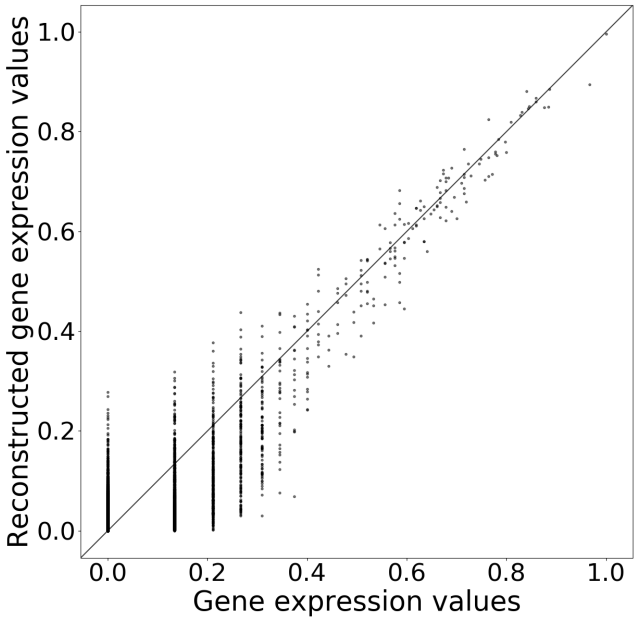
mean). To determine whether this decrease in performance is caused by a bad combination of activation functions used in the deep autoencoder, I tested different combinations of the ReLU6 activation function in the decoder output layer with other upstream activation functions. I've picked two best models identified in the earlier comparison, see Table 3.2, and trained them using the ReLU6 activation function in the decoder output layer. The performance of the model with the Tanh activation in the hidden layers and the ELU activation in the encoder latent layer became worse - the reconstruction error increased from 27.3830 to 27.7511. The performance of the model with the Tanh activation in the hidden layers and the Softsign activation in the encoder latent also became worse - the reconstruction error increased from 27.4378 to 27.6197. These results suggest that increasing the range of the data does not enable the deep autoencoder to learn more efficiently.

The majority of the values in the data being equal to zero means that the weights in the encoder hidden layer are mostly multiplied by zeros. If a product of a value in the data and a weight is zero it provides no gradient information for adjusting this weight throughout training. Since a single bias is added to each nodes value, the bias cannot accomodate for the lack of informative gradients for numerous genes. This explains why accommodating for zero values in the data by using the ReLU activation function in the decoder output layer did not have a positive effect on the performance. Another way to accommodate for zeros in the data and the lack of information they provide is to shift the range of the data. I scaled the data to [0,2] range and shifted it to [-1,1] range. Now all of the values in the data are non-zero

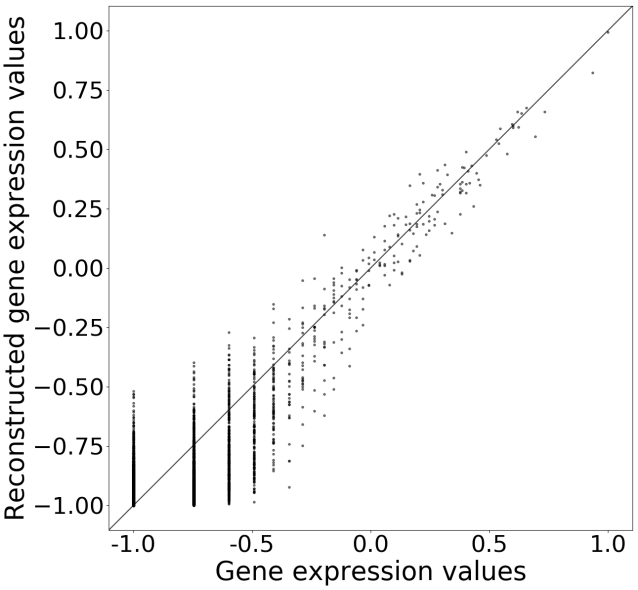
and as they get multiplied by weights they provide an informative gradient that allows to adjust the weights throughout training. There are two choices for the decoder output layer activation function - Tanh and Softsign. To be able to compare the performance of this model the reconstruction errors were computed on the data in the original [0,1] range. The deep autoencoder with the Softsign activation function in the decoder output layer trained on the data in [-1,1] range performs worse than the original autoencoder - the reconstruction error increases from 27.4840 to 28.0226. If the Tanh activation function is used instead the reconstruction error decreases from 27.4840 to 27.3948, which is not a significant difference (the value is less than 2 standard deviations away from the mean of the distribution). Figure 4.3b reveals that scaling the data to a different range and shifting the dropout values to -1 instead of 0 does not solve the underlying problem. The deep autoencoder is able to reconstruct high expression values in the data (the points in the upper right part of the plot show strong correlation between real and reconstructed gene expression values), but the low expression values are dominated by noise that the model is not able, and not expected, to reconstruct (the points in the lower left part of the plot show lack of correlation).

4.1.3 Accounting for dropouts in the data

A large proportion of zeros in scRNA-seq data correspond to technical dropouts. This property of the data is often compared to the problem of denoising an image with many pixel values set to 0. The ideas from the Vincent et al. [2008] paper



(a) The original autoencoder.



(b) The modified autoencoder.

Figure 4.3: (a) The expression values in a cell plotted against the values reconstructed by the original autoencoder. (b) The expression values in a cell plotted against the values reconstructed by the deep autoencoder with the Tanh activation function in the decoder output layer trained on the data in [-1,1] range.

that coined the term denoising autoencoder have been applied to scRNA-seq data in numerous studies [TAN et al., 2014, Deng et al., 2019, Amodio et al., 2017]. Denoising autoencoders perform well on the types of data where the values are spatially or temporally linked, like images and audio. This is not the case for scRNA-seq data where genes in the expression vector are ordered randomly or alphabetically. Ordering the genes by their position on a chromosome does not lead to a data where values located next to each other are more likely to contain information about each other. Similarly, clustering genes by their annotated function would not resolve the problem of how to arrange the clusters between themselves and how to arrange the genes inside the clusters. The gene interaction network inevitably has much more irreducible complexity than the interaction network of pixels in an image, where the amount of information about a pixel contained in other pixels is a simple monotonically decreasing function of the distance between them. Hence, studies that use autoencoders for dropout imputation in scRNA-seq data are of limited utility.

An alternative way to deal with this property of the data is to consider the parallel between the the technical dropouts in scRNA-seq data and a (confusingly called the same way) dropout layer in a neural network. The idea behind a dropout layer is simple - randomly deleting some of the values in the data (for example, setting some pixel values to zero in an image) leads to a more robust neural network that cannot rely on any specific value in the data and instead has to learn robust features that are based on an interplay between many different values with no single one

having an overwhelming importance. The concept of a dropout layer has been introduced by [Srivastava et al., 2014] and has been used extensively since then to prevent overfitting in neural network training. Khalfaoui and Vert [2018] were the first to suggest a parallel between the dropouts in scRNA-seq data and a droupout layer in a neural network.

In this section I will explore the effects of introducing a dropout layer upstream from the encoder in the deep autoencoder, see Figure 4.4. I expect that the dropout layer will lead to two direct benefits. First, it will force the deep autoencoder to learn features that are robust to missing values. Second, it will limit the overfitting since the values are randomly deleted from the data at the start of every epoch of training and hence the model never sees the same expression profile twice. Arpit et al. [2017] showed that dropouts can hinder memorization in GNNs while preserving their ability to learn from real data. I implemented the dropout layer in the following way - for every non-zero value in an expression profile a random number in the [0,1] range is generated, if the expression value is less than the randomly generated number then the expression value is set to zero. This way the probability of a gene becoming a dropout is proportional to the level of expression thus crudely mimicking the properties of the data. The number of zeros in the data after the dropout layer increases from 87.64% to 97.74%. This means that the deep autoencoder trained on this data will be more robust to the technical dropouts in the data, since this additional layer exaggerates the problem in a realistic manner.

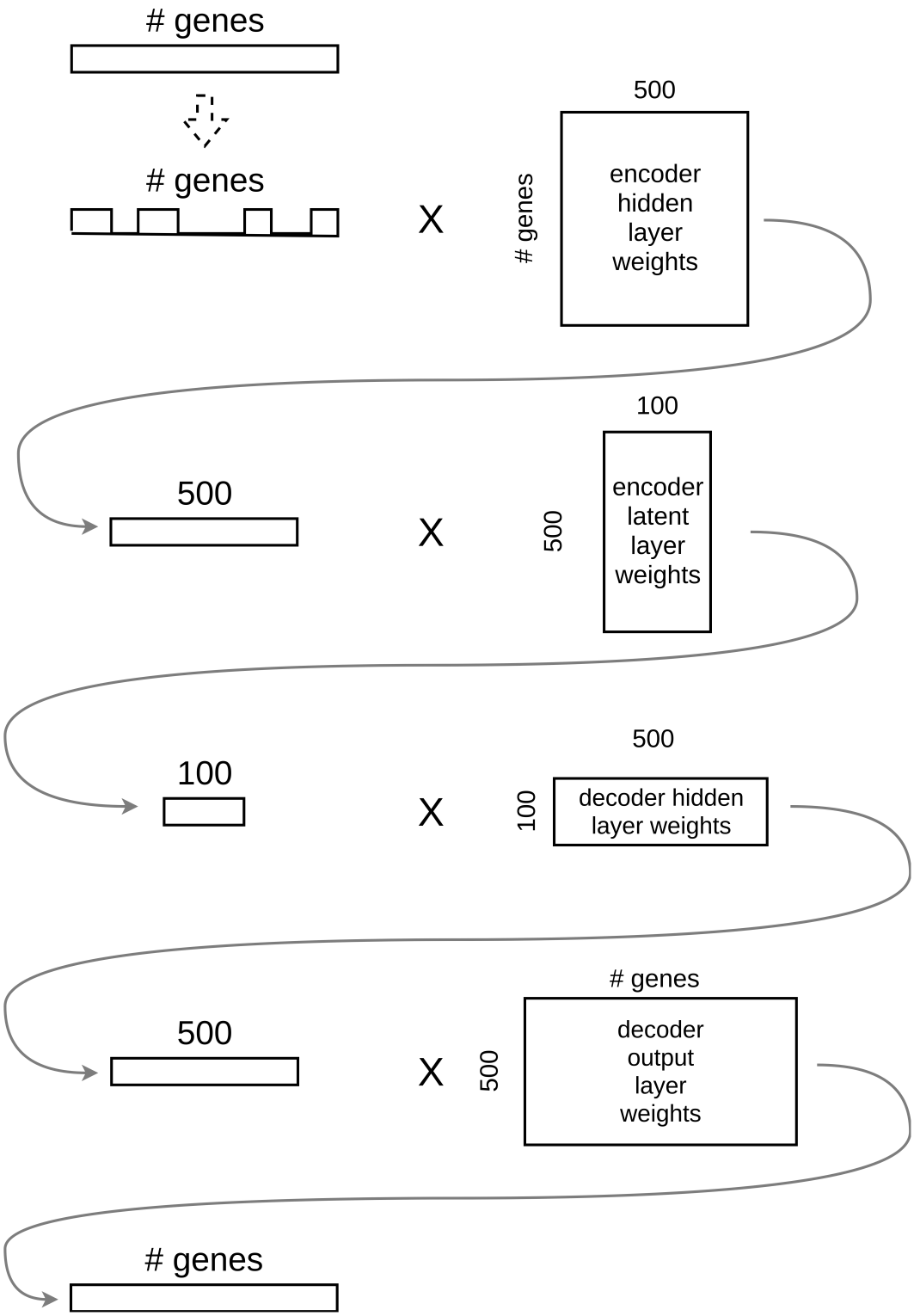


Figure 4.4: A deep autoencoder with a dropout layer upstream of the encoder.

The denoising autoencoders train by encoding a noisy images, decoding it and aiming to minimise the reconstruction error between the output and the target noise-free image. This approach does not work for scRNA-seq data as the target noise-free true gene expression profile, i.e. ground truth, is not available. It is unrealistic to expect that an autoencoder will be able to distinguish between the dropout values that occurred during the experiment and the artificially introduced dropouts. To confirm this, I trained the deep autoencoder with a dropout layer upstream from the encoder in a usual manner - comparing the reconstructed gene expression profile with the original data (before going through the dropout layer). The model did train, but, as expected, the performance was very poor - the reconstruction error equal to 165.4284.

It is unrealistic to expect that an autoencoder will be able to distinguish between real and artificially introduced dropouts, or that it will be able to infer the true unobserved expression values for dropouts present in the data but not the noise-free levels of expression of other genes. Therefore, I used an approach that has not been explored before. I trained the deep autoencoder by comparing the reconstructed gene expression profile with the output of the dropout layer instead of the original input data. The deep autoencoder trained in this way is not expected to be able to impute the technical and/or artificially introduced dropout values. Instead, the aim is to able to learn features from the imperfect data. The reconstruction error of this autoencoder on real data is 36.5603, which implies that this is a promising direction to explore compared to conventional approaches that train the autoencoder with a

dropout layer by comparing to the data prior to the dropout layer.

4.2 Biologically inspired GNN architectures

The deep autoencoder takes gene expression values as input, combines them in a non-linear manner into a set of 500 features, which are subsequently combined into 100 latent features. The process of the information flow in a living cell is much more intricate, and it also includes information from outside the cell. While the input into an autoencoder, in this current implementation, is limited to gene expression values only, the internal connections in the deep autoencoder can be modified to create a more realistic model of the information flow and thus produce more informative latent representations of the cell states. In this section I will explore two modifications - introducing additional hidden layers and residual connections.

4.2.1 Deeper autoencoders

The gene expression levels in a cell result from an interplay of regulatory mechanisms, which are in turn regulated through various biological pathways. Combinations of these pathways result in biological processes which collectively create the global signalling network of each individual cell. All of this hierarchical information is observed through a narrow window of a single expression profile. In this profile each gene's expression might be a result of being regulated through numerous pathways associated with different processes all happening simultaneously in a cell. To capture this hierarchical information I will introduce additional hidden

layers to the deep autoencoder. Increasing the number of hidden layer in both the encoder and the decoder to two results in the reconstruction error equal to 27.3673, increasing it to three further decreases the reconstruction error to 27.3273. The improvement is not significant.

I examined the performance of the autoencoder with three hidden layers both in the encoder and the decoder. In this model the latent space representations of the data points are interpreted as cell states that are combinations of biological processes occurring in the cell (the downstream hidden layer). In turn, these processes correspond to one or more active or repressed pathways (the next downstream layer) which involve several transcription factors (the next downstream layer). The activity of these transcription factors ultimately result in the gene expression levels in a cell (the output layer). Figure 4.5 shows that most of the layers in the encoder actively learn certain features, they do not simply pass down the information from the upstream layer, which would result in maximum correlations to features in the upstream layer approaching 1. The trained model uses all of the 100 latent dimensions, Figure 4.6, which was not the case for the original autotoencoder that only used 93 of them. This implies that the performance of this model can be improved by increasing the number of nodes in some or all of the layers.

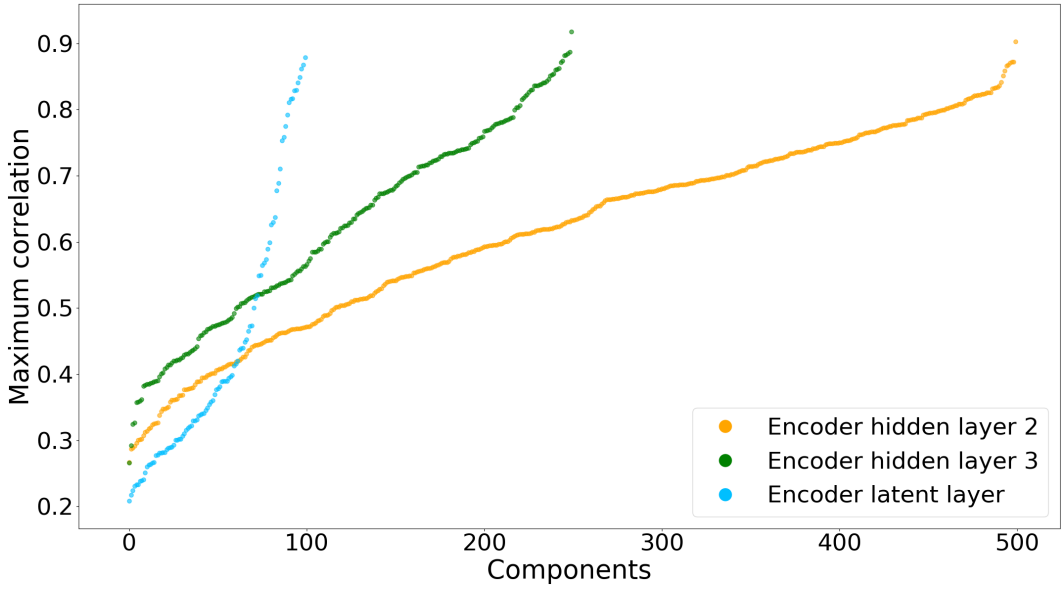


Figure 4.5: The maximum correlations between the features in the encoder layer and the features in the layer directly upstream from it.

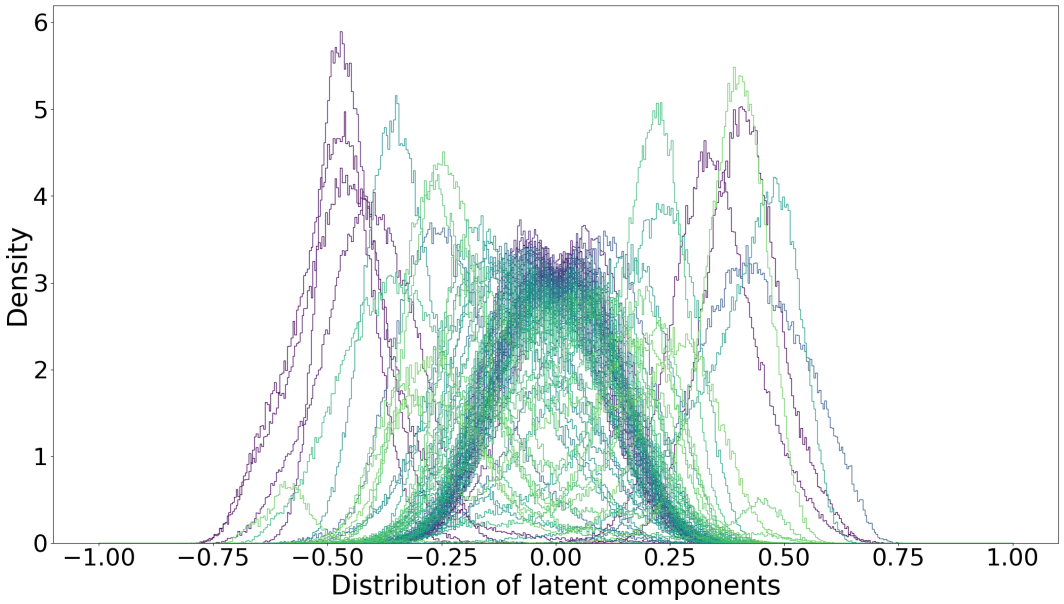


Figure 4.6: The distributions of 100 latent components. Each of the components is shown in a different colour simply for ease of visualisation.

4.2.2 Residual connections

In the original autoencoder a latent space embedding of a cell is created using the information contained in the features of the encoder hidden layer, which in turn is based on the information stored in the expression profile. This strictly sequential flow of information is biologically unrealistic. The nodes of the encoder hidden layer have access only to the gene expression values, other information about the identity of the cell or the processes occurring in the cell is not available. The idea is that the autoencoder can use the information contained in the expression profile to infer this additional information, i.e. create features corresponding to the cell's identity and activities. Here is the problem: nodes in the latent layer are connected only to the nodes in the hidden layer, and hence they do not have direct access to the information about gene expressions. In a real cell the information about cell's identity, however it is encoded, is combined with the information contained in the gene expression levels to orchestrate cell's development and activities. See Stadhouders et al. [2019] for a good explanation of how a cell state can be interpreted as an "emergent property" that arises from the interplay of various components that govern transcriptional regulation in a complex, multi-layered and interconnected manner. Stadhouders et al. [2019] review recent studies suggesting that the interplay between transcription and genome conformation governs cell-fate decisions, i.e. that both transcriptomic and non-transcriptomic information is utilised at the same time. Another example where the information about the gene's expression level is used alongside with cell state information is cell-fate-instructive transcription factors - they can short-circuit signal transduction processes when

over-expressed, which results in a complete rewiring of a cells expression programs [Graf and Enver, 2009].

To create a model that is able to capture this flow of information between the different hierarchical levels in a cell, I used residual connections. This powerful concept was introduced by He et al. [2015b], and it lead to an enormous progress in image recognition in just a couple of months [Szegedy et al., 2016]. A deep autoencoder with residual connections uses the information in an expression profile to create the encoder hidden layer features, and then it uses both these features and the expression profile to create a latent representation of a cell, see Figure 4.7. The reconstruction error produced by this model is equal to 27.4607, which is similar to the original autoencoder (within 0.5 of a standard deviation). I examined the weights in the encoder hidden layer to estimate how much importance a trained model places on the available information about the expression values compared to the features from the upstream layer. Out of 100 nodes, in 64 the highest weight is associated with a feature and in 36 with a gene, see Figure 4.8. The nodes with higher maximum weigh values are more likely to have those values associated with a feature. From this I conclude that features learnt by the upstream layer are more "informative" than gene expression values on average, as expected. At the same time some of the gene expression values provide essential information not in the context of expression values of other genes but in the context of the features learnt by the hidden layer. Amongst the genes that have a major impact on one of the 36 nodes dominated by the transcriptomic information there are ID2 and ID3 inhibitors

of DNA binding, BTG2 anti-proliferation factor, HSPA1A heat shock protein and, as expected, numerous transcription factors (CEBPD, IRF8, TSC22D1, HES1, etc.). Similarly, the most negative weight is associated with a feature in 58 nodes. This implies that both types of information are encoded using both linear and non-linear domain. Notably, there are two genes that are associated with by far the highest weights in this layer, both of them are members of the keratin gene family (KRT13 and KRT17) that are highly expressed in skin tissue. To assess the influence of available transcriptomic information on the overall behaviour of this layer, I examined the distributions of positive and negative weights associated with each type of information, see Figure 4.9. The features created by the encoder hidden layer undoubtedly play a mayor role (i.e. are associated with larger weights), but the information contained in the gene expression values is also valuable.

4.3 Shaping the autoencoder training process

4.3.1 Shaping feature allocation to nodes

An autoencoder with sufficient capacity, i.e. number of nodes, to model the data has no incentive to allocate individual features to specific nodes. The training is governed solely by the loss function that measures the quality of the expression profile reconstruction. This results in partitioning of concepts that a biologist would regard as a single feature, for example a cell cycle phase, across several dimensions in the latent space. In Chapter 3 I showed that 5 out of 100 latent dimensions trained on the Cheng et al. [2018] dataset were related to cell cycle phase, and more than

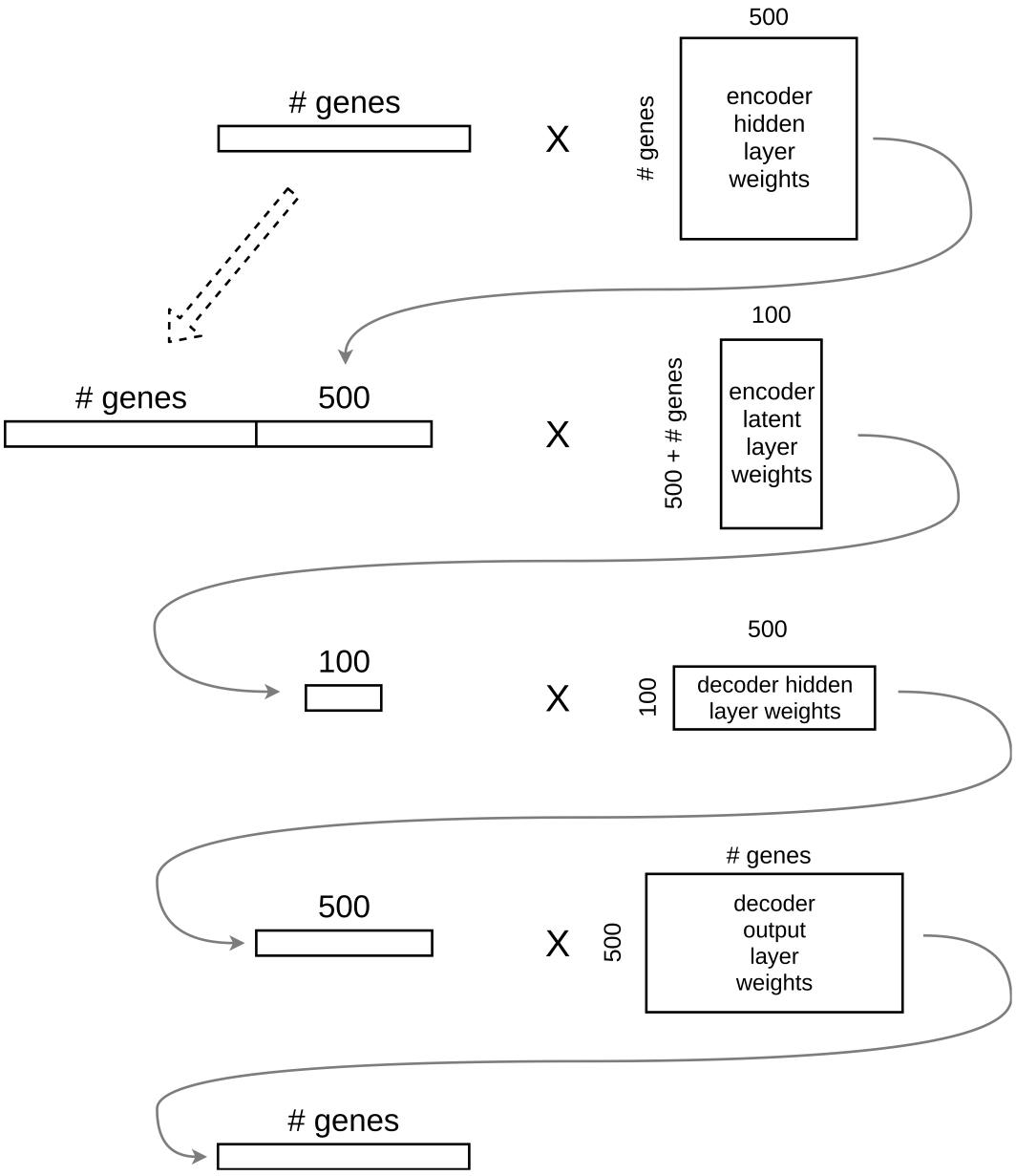


Figure 4.7: A deep autoencoder with residual connections uses the information in an expression profile to create the encoder hidden layer features, and then it uses both these features and the expression profile to a create latent representation of a cell.

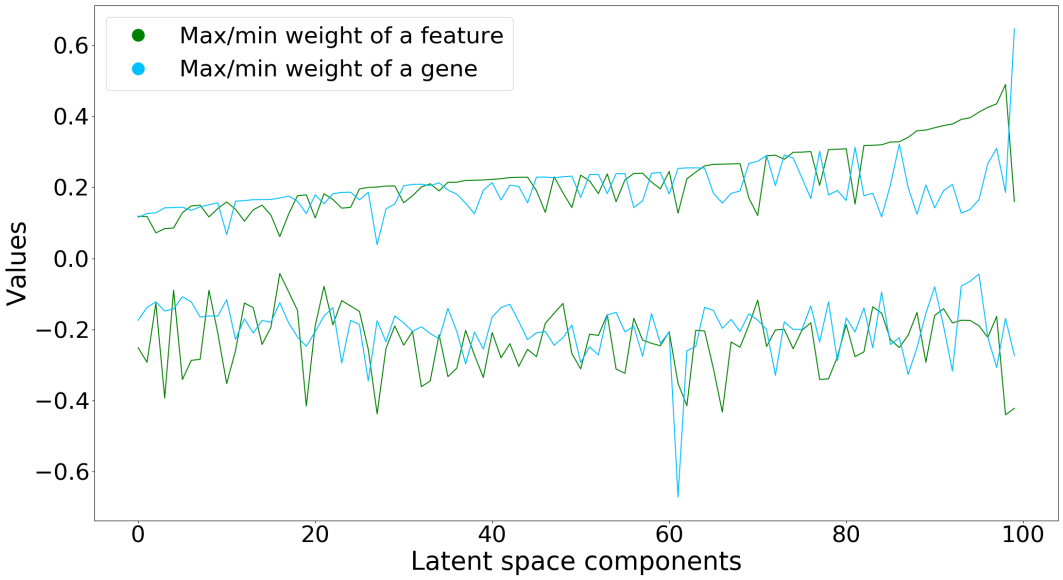


Figure 4.8: Maximum and minimum weights associated with the gene expression values (in blue) and with the features from the upstream layer (in green) for each of the nodes in latent layer.

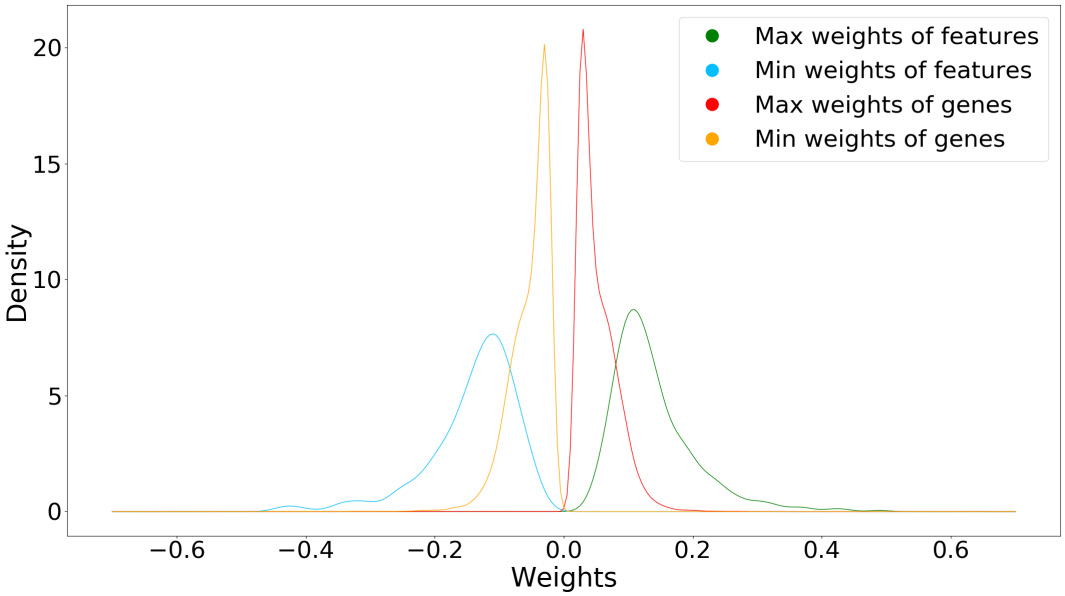


Figure 4.9: The distributions of positive and negative weights associated with the gene expressions values and the features from the upstream layer.

8 contained some batch effects. To address this biologically meaningful feature partitioning across nodes, I used the approach suggested by Ladjal et al. [2019]. First, I trained an autoencoder with 20 nodes in the encoder hidden layer, 4 nodes in the encoder latent layer and the original decoder with 500 nodes in the hidden layer, see Figure 4.10. Next, I expanded the dimensionality of the encoder by adding 15 more nodes to the hidden layer and 3 more to the latent layer and kept the trained weights from the previous round of training in both layers of the encoder. Then, I trained the expanded autoencoder with a mixture of newly initiated and trained weights in the encoder and only newly initiated weights in the decoder. I kept adding nodes to both layers of the encoder until the desired dimensionality was reached. Ladjal et al. [2019] showed that, in applications to image based problems, an autoencoder trained by starting with a very low dimensionality and adding small number of additional nodes throughout training produces a latent space constituted of independent components ordered by decreasing importance. I examined whether these PCA-like properties could be achieved outside the domain of image based applications.

The properties of the latent space created by the autoencoder trained in a dimensionality expanding manner are much different from the intended ones. One of the four initial components remained unused and the others had a uni-modal distribution centred at 0. All three of the components introduced during the first round of expansion remained unused. Only starting from the second round of expansion the components started to capture features of the data. Triplets of latent

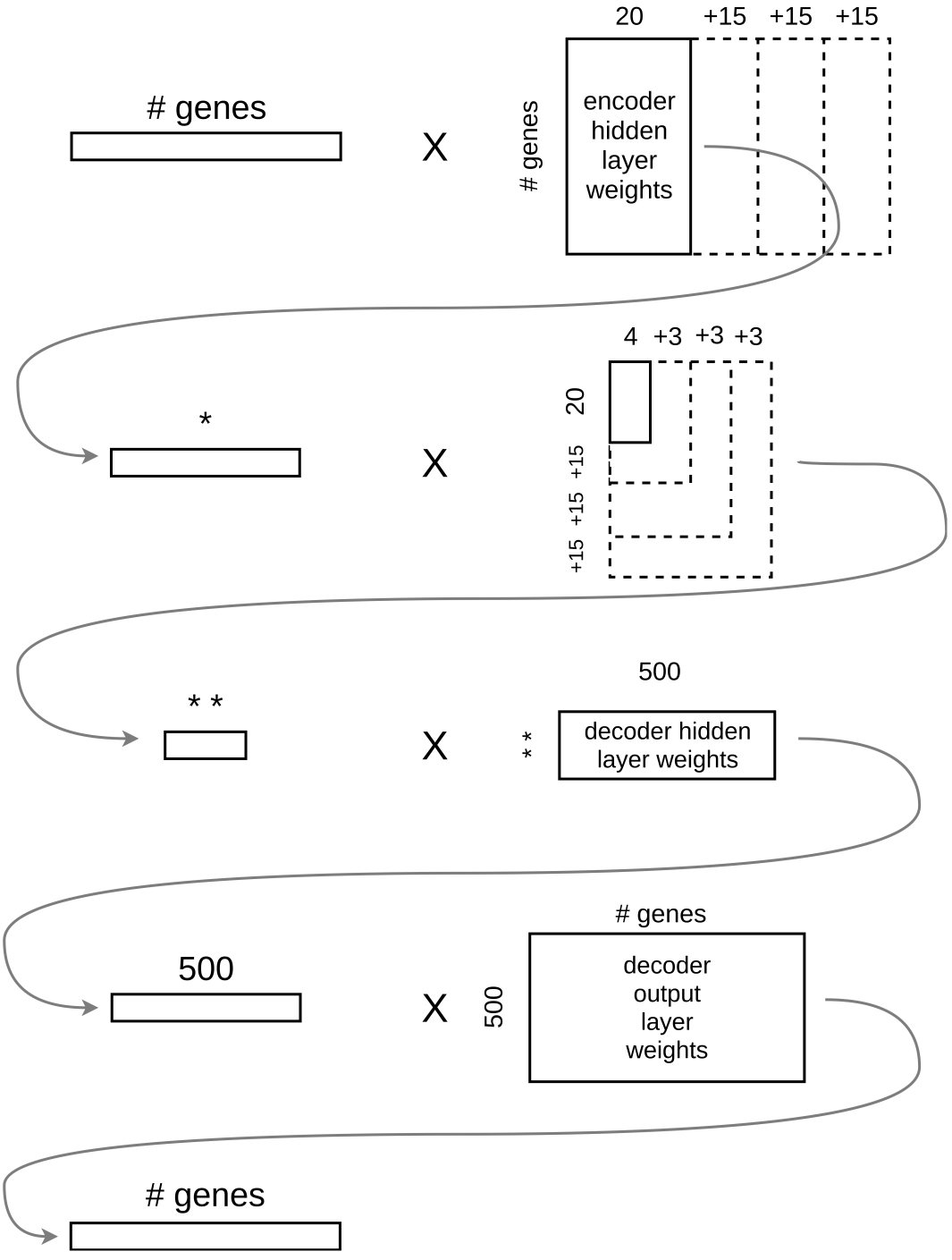
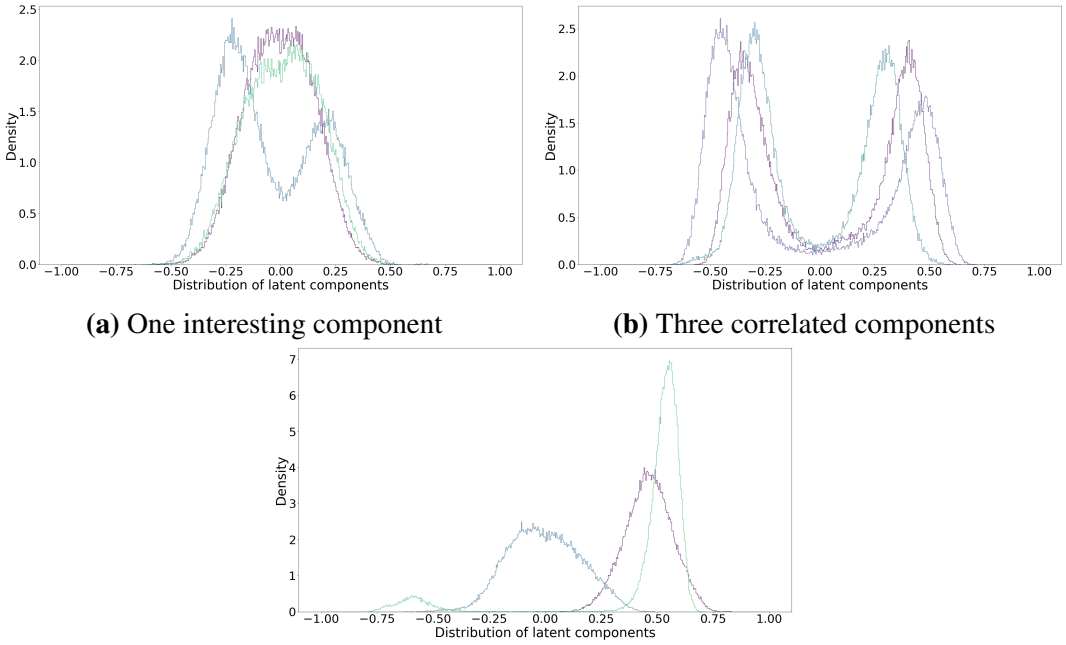


Figure 4.10: An autoencoder trained by starting with a very low dimensionality and adding small number of additional nodes throughout training.

nodes corresponding to different rounds of expansion show different patterns - a single component with an interesting distribution and another two with a standard uni-modal distribution centred at 0, three correlated components with a distinctly shaped distribution, each of the three components with a different distribution. Figure 4.11 shows some of these examples. It is interesting that the components that were present in the network from the first two rounds of training, and initially failed to train, never retrained from that state. Contrary to the expectations, 11 of the latent components in this model correspond to batch effects, and the cell cycle effect is partitioned across several nodes that were added at different rounds of expansion. To investigate whether the failure to identify important features was caused by insufficient number of nodes, I repeated the same training procedure by initiating the network with 10 nodes in the latent layer instead of 4. The dynamics remained largely unchanged - some of the initial components failed to train, while batch and cell effects remained split across numerous components.

4.3.2 Pre-training individual layers

The lack of significant improvement resulting from adding the third hidden layer that we observed in Section 4.2.1 might be linked to the lack of gradients propagating through all the layers of the encoder. To allow the network to train more efficiently, Larochelle et al. [2009] suggested pre-training shallow neural networks with appropriate number of dimensions and subsequently stacking them into a deeper architecture. This approach can adopted for this application, see Figure



(a) One interesting component

(b) Three correlated components

(c) Three unique components

Figure 4.11: Triplets of latent nodes corresponding to different rounds of expansion show different patterns - (a) a single component with an interesting distribution and another two with a standard uni-modal distribution centred at 0, (b) three correlated components with a distinctly shaped distribution, (c) each of the three components with a differently shaped distribution. In each plot the components are shown in a different colour simply for ease of visualisation.

4.12. I first trained 3 autoencoders with encoder hidden layer sizes 1000, 500 and 250, while keeping the decoder hidden layer size constant at 500. Then I stacked the weights from pre-trained encoder layers into a deeper autoencoder and trained the model to optimality. The reconstruction error produced by this model is equal to 27.7200, significantly worse performance compared to the original autoencoder (more than 5 standard deviations from the mean). To improve on this, I used symmetrical autoencoders for pre-training the layers instead of keeping the decoder hidden layer constant. This allowed the individual models to train better and hence resulted in the improved reconstruction error 27.4625, which is similar to the performance of the original autoencoder (within 0.5 of standard deviation from the mean).

4.4 Reproducible autoencoder training

4.4.1 Autoencoder training consistency

The deep autoencoder is an over-parametrised model, and hence if it is trained more than once on the same data the results will be different. In this section I will examine the properties and magnitudes of these differences. To do so I trained the deep autoencoder on the Cheng et al. [2018] dataset 20 times. The training dynamics is very similar across all runs. The reconstruction quality of the test set after the first epoch varies, but there is no relationship between the success of the first epoch and the performance of the model trained to optimality (Pearson correlation coefficient equal to 0.1011). Comparing the latent features between two training runs shows that the latent representations they created are different, it is not just the case of permuting the features to find approximate one-to-one correspondence. The maximum correlations between latent features created by one of the training runs and the features created by a different run ranges from 0.1374 to 0.9956, see Figure 4.13. The average correlation between one feature and all the features produced by another training run is low, see Figure 4.14. The reconstruction errors for each individual cell are highly correlated - across the 20 training runs the lowest correlation between a pair of runs is 0.9996. This implies that there exist some underlying properties that determine whether an autoencoder will be able to reconstruct a particular expression profile after training on a particular dataset.

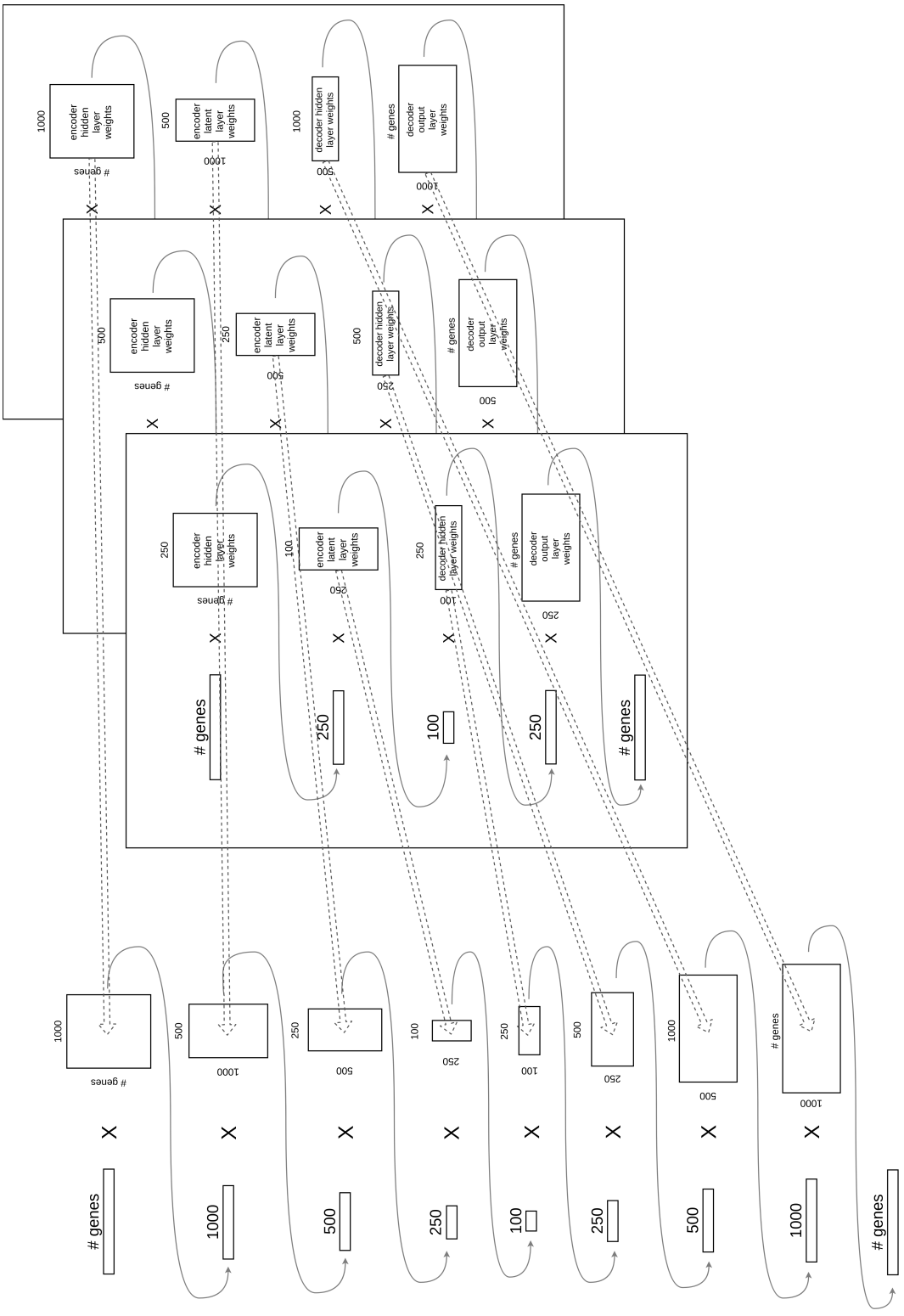


Figure 4.12: Enabling the training of deeper architectures by pre-training the weights in shallow architectures.

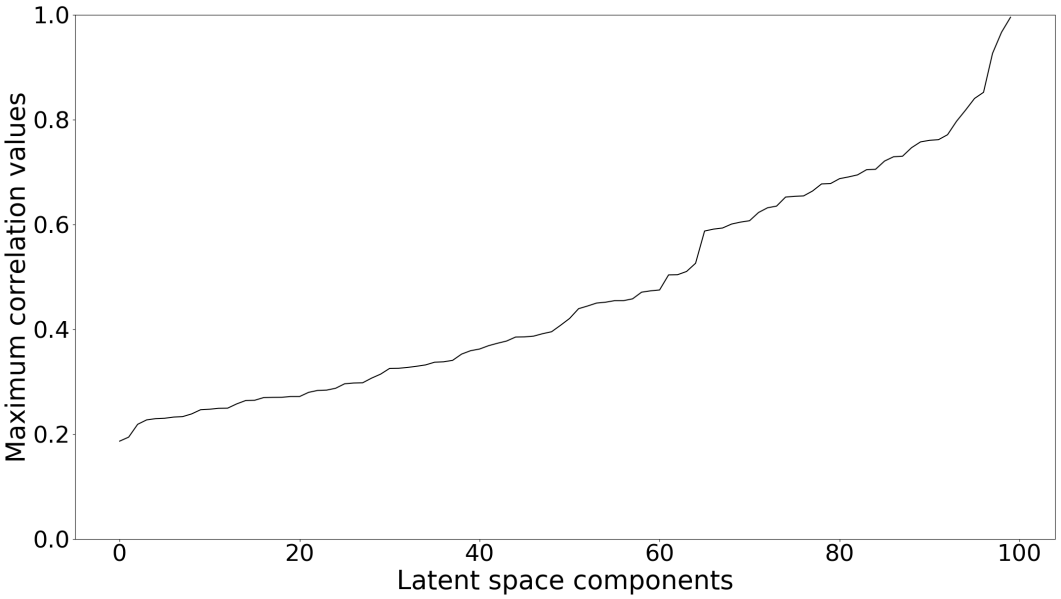


Figure 4.13: The maximum correlations between the latent features created by one of the training runs and the features created by a different run.

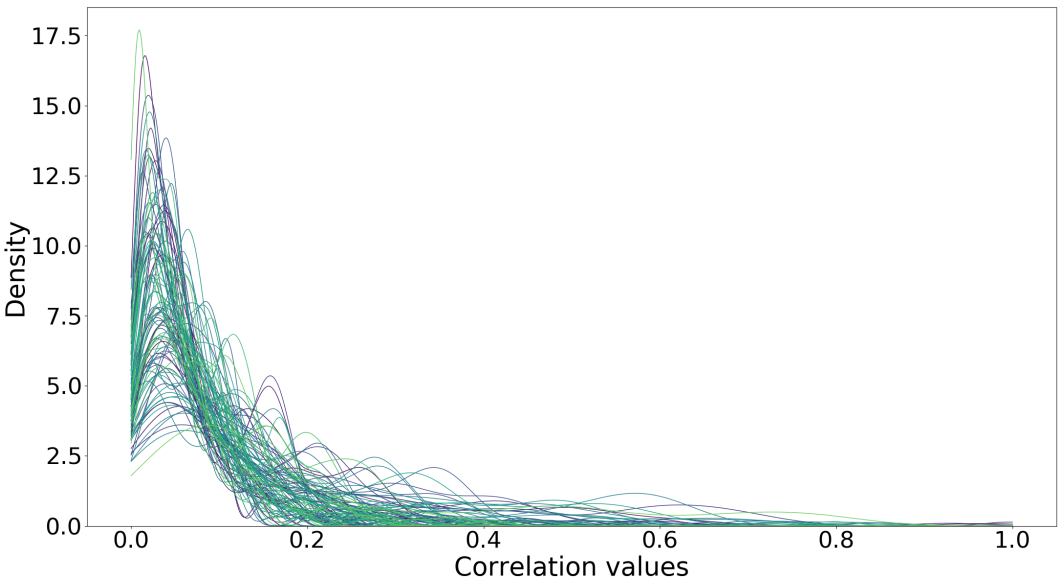


Figure 4.14: The distributions of correlations between a latent feature and all of the features produced by another training run. Each distribution is shown in a different colour simply for ease of visualisation.

4.4.2 Reproducible autoencoder

To address the lack of reproducibility I propose the following strategy, see Figure 4.15. First, the original autoencoder is trained repeatedly (for example 10 times) on the same dataset. Then, all 10 encoder hidden layer weight matrices are combined into one ten times bigger layer. This autoencoder with a ballooned encoder hidden layer is trained once. Based on the resulting values in the weight matrix in the downstream layer, 500 "most useful" of the 5000 features are selected. The autoencoder with 500 nodes and a corresponding pre-trained weight matrix in the hidden layer is again trained 10 times. Similarly to the strategy for the hidden layer, all 10 latent layer weight matrices are combined into one ten times bigger layer. This autoencoder with a ballooned latent layer is trained once. Based on the resulting values in the weight matrix in the decoder hidden layer, 100 "most useful" of the 1000 latent dimensions are selected.

Performing the steps described above twice, will allow the comparison between the two resulting latent space embeddings. If one-to-one correspondence between the components of the two latent spaces can be established, that would imply that the strategy has been successful and a meaningful analysis of the latent space can be carried out. If such correspondence can not be established this could be a result of one of the two underlying causes. Either 10 training runs were insufficient to capture all the diversity of the features that can be learnt from the data, or the network architecture is not suitable to learn efficiently from this particular dataset. In the former case, simply increasing the amount of training runs for generating

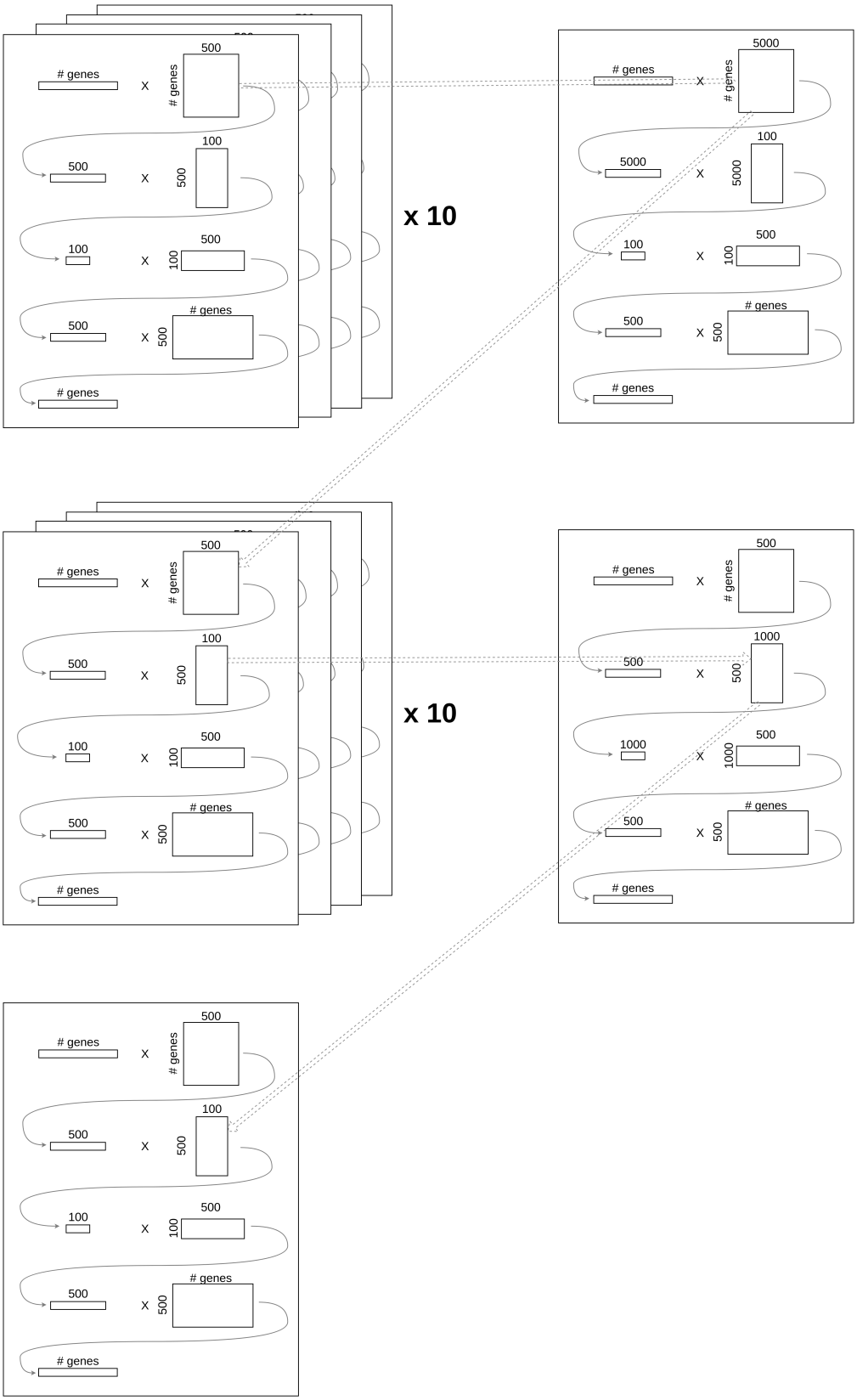


Figure 4.15: A more reproducible autoencoder.

the features of both the encoder hidden layer and the decoder layer will result in a reproducible autoencoder. This does come at a cost though, since the amount of computation will increase at every step of the training process. In the latter case, the inability to create a reproducible autoencoder is a useful indication that an architectural change is required for a successful application of an autoencoder to this specific dataset. It is, however, not possible to distinguish between the two cases in any other way apart from first trying to improve the reproducibility by increasing the number of training runs.

4.4.3 Conclusions

In this chapter I've explored the opportunities to improve upon each component of the process of applying (GNNs) to the scRNA-seq data. In terms of data used for training a GNN, I've shown that while more data is always better ultimately every experiment contains a finite amount of information. Adding more data from the same experiment results in ever diminish returns. I've explored several strategies for transforming the data to make it more suitable for a neural network training and to tailor it to a specific loss function used. This strategies were not successful. I've shown that a dropout layer applied correctly is a promising direction to explore. Tailoring the probability function that determines the number and the properties of the additional dropouts introduced based on the properties of a specific dataset is expected to result in a more robust model with improved performance. After considering all component of the process of applying (GNNs) to the scRNA-seq

data, I conclude that developing biologically inspired GNN architectures (combining deeper autoencoders with residual connections and other ways to incorporate hierarchical information) to mimic information flow in a cell is the most promising direction. The performance of GNNs applied to scRNA-seq data can be improved in two important directions - reproducibility and biologically meaningful interpretability.

Chapter 5

Discussion

In the first section of this chapter I will summarise the contributions of my work and my conclusions based on the results of my work. This will be followed by putting my work into the context of current efforts aimed at applying neural networks to scRNA-seq data. In the second part of the chapter I will offer my perspective on the future of single cell techniques and the analysis methods associated with them. In particular, I will explain the need for a theoretical framework around the concept of a cell type, I will cover the new experimental techniques that measure multiple modalities in the same cell and techniques that capture temporal dynamics of transcription. I will highlight how these advancements in the types and volumes of single cell data available result in a need for machine learning based methods which are able to extract the information from the data that would not be available with statistical methods.

5.1 Assessment of GNN applications to scRNA-seq data

In the Introduction chapter of this thesis I've covered the properties of the scRNA-seq data and the intricate structure of noise present in this data. In an comprehensive survey of the different types of methods that have been developed specifically for scRNA-seq data or re-purposed to be applicable to it, I explored how do these methods fit in with the demands of different scientific questions that scRNA-seq data aims to answer. This motivated the need for new methods that would address issues that are currently not widely acknowledged because they bring into question the validity of commonly used analysis methods and previously published results. In this section I will summarise my contributions to this field, before putting them into a wider context of how the field has developed during the time of this project being carried out.

One part of my work was focused on the assessment of the potential of autoencoders to be adopted as a useful and flexible tool for scRNA-seq data analysis. Unlike all other methods that create a GNN-based model and compare its performance with PCA as a benchmark, I started by showing the mathematical equivalence between PCA and a shallow linear autoencoder. I then added complexity to the model, step by step, until I reached a deep autoencoder, thus demonstrating the link between the familiar method and the newly adopted GNNs. I explored the information flow through the network with the aim to provide a better understanding of what are the

current capacities and limitations of an autoencoder with a basic architecture. I examined the properties of the latent space created by the autoencoder. My conclusion is that interpolation and extrapolation opportunities that attract much attention are not feasible. This is mainly due to the fact that these methods would require several strong assumptions about the the data - an assumption that the latent space could be populated uniformly if enough relevant data is collected and an assumption that the properties of data points vary smoothly in all dimensions along the interpolation trajectory - neither of which is true.

Most of the currently employed scRNA-seq data analysis methods are based on assumptions about the data that are either not testable or do not hold, at least not in general. The opportunities provided by the progress in machine learning are readily adopted, as are their assumptions about the data. Instead of extending the existing arsenal of methods that are confounded by the assumptions that the scRNA-seq data cannot satisfy, I advocate for taking this into a different direction. The GNNs are unique in a sense that they learn from reproducing the data and hence there is an inherent opportunity to assess their performance without any additional knowledge about the data. By exploiting the flexibility of neural networks it is possible to design biologically meaningful architectures of autoencoders and use these to create useful embeddings of the data into lower dimensional space. Hand-picking the relevant combinations from the constructed latent space can be used both for visualising the aspects of the data that are of interest and for designing a distance metric that would be uniquely suitable for a particular application.

Another part of my work explores how alterations to the data, the autoencoder's internal structure and the training dynamics can positively influence the performance and the interpretability of the autoencoder. I advocate for more work on designing GNN architectures that are informed by our understanding of biology. Based on my results I conclude that predicting an interaction of biological data with a neural network architecture, neither of which we understand fully, is not always possible and hence more effort is required at two fronts. First, innovative ideas of how to transform scRNA-seq data into a format more amenable to neural network training are required. Second, dedicated work aimed at making progress in machine learning applied to tabular data is essential. Cutting edge machine learning research is focused on image-based problems and natural language processing, while the contributions of machine learning to biology are sparse. More resources invested in machine learning applications to tabular data would revolutionise not only biology but many other fields too.

My conclusion is that GNNs are a useful tool for scRNA-seq data analysis. They are good at identifying prominent features in the data and producing a lower dimensional embedding of the data. This embedding can be then used both for data visualisation purposes and for designing a meaningful distance metric based on hand-picked latent dimensions. Looking at the distributions of particular latent dimensions also allows to distinguish between cell types (for example, melanocytes or immune cells in the skin tissue) and transitions between them that look distinct

(similar to a rare cell type) but are cell stages and not types. Two important advantages of GNNs are their flexibility and scalability. Unlike PCA that requires the whole dataset to be loaded into computer memory to compute the PCs, GNNs require only a small batch of data at a time. This means that there is no practical limit on the amount of data used for training a GNN. The time required for a single epoch of training grows linearly with the amount of data, and given more data less epochs of training might be required (depending on the heterogeneity of the data). Another advantage is that GNNs are good at ignoring irrelevant variables and hence there is no need for identifying a subset of variables that should be used for an analysis. Unfortunately, GNNs have no magic superpowers - they learn from the information present in the data, but they are not able to create new information. The property of scRNA-seq data is that expression level estimates for genes expressed at low levels are noisier that the ones for highly expressed genes. As a result, GNNs are good at encoding and decoding the expression values of highly expressed genes, but they perform poorly at reconstructing the low expression values (the information is simply not present in the data). Similarly to other methods, GNNs are not currently suitable for interpolation or extrapolation in the data, but unlike other methods GNNs hold the most potential in this area. The flexibility of GNNs is one of the main reasons for future research into their applications to scRNA-seq data analysis. Their architecture can be designed based on our understanding of biology and specifically information flow in a cell. External clues like chemical environment and signals from neighbouring cells can be incorporated by using conditional GNNs or additional input layers.

5.2 How variational autoencoders became popular

In 2014 Casey Greene pioneered the application of autoencoders to gene expression data [TAN et al., 2014]. They further explored this approach with ADAGE, eADAGE and Tybalt [Tan et al., 2016, 2017, Way and Greene, 2017]. Their work highlighted the existing opportunities in this field, but conventional methods of data analysis prevailed and there was not much progress in this field apart from a couple of exploratory papers [Chen et al., 2016, Barsacchi et al., 2018]. The scRNA-seq data becoming widespread lead to an immediate avalanche of 14 papers published in the period from October 2018 to March 2019: VASC [Wang and Gu, 2018], scVI from Nir Yosef lab [Lopez et al., 2018], SSCVA [Gold et al., 2018], scGen from Fabian J. Theis lab [Lotfollahi et al., 2019], GSAE [Chen et al., 2018], Deep count autoencoder (DCA) again from Fabian J. Theis lab [Eraslan et al., 2019], scVAE from Ole Winther [Grønbech et al., 2018], SAUCIE from Smita Krishnaswamy lab [Amodio et al., 2017], DESC [Li et al., 2019b], [Fan et al., 2019], CIC [Abdolhosseini et al., 2019], Dhaka [Rashid et al., 2019], scScope from Altschuler and Wu lab [Deng et al., 2019] and Dr.VAE [Rampášek et al., 2019]. This avalange was followed by a continuous stream of additional methods that apply GNNs to scRNA-seq data: Cyclum [Liang et al., 2019], SISUA Trong et al. [2019], Targonski et al. [2019], [Kinalis et al., 2019], scAlign [Johansen and Quon, 2019] and most recently CVA [Lukassen et al., 2019]. The frequency of the new methods being published is indicative of none of them being adopted for general use. Each of these methods are

build on an interesting idea, mostly centred around pre-processing the data prior to using it for training or using a trained GNN in a creative way. There has been hardly any progress in terms of engineering GNN architectures motivated by the biological knowledge, shaping the learning process of a GNN to increase it's applicability to this specific application, or addressing the unreproducibility of GNN training. The work I presented in Sections 4.2, 4.3 and 4.4 remains novel and addresses the issues that have not yet received enough attention.

5.3 Future of single cell techniques

The future of biological research at single cell level lies both in design of new experimental protocols and in development of useful theoretical concepts. In this section I will discuss recent developments of experimental techniques with a particular focus on recording more than one type of data from a single cell and quantifying the temporal dynamics of a transcriptional cell state. Prior to that, I would like to highlight the need for theoretical concepts that were not required before the widespread use of single cell techniques and are hence not well developed. Svensson and da Veiga Beltrame [2019] demonstrated that the number of cell types identified in scRNA-seq studies is proportional to the number of cells captured by those experiments. This observation shows that the current working definition of a cell type is closely linked to clusters produced by a clustering algorithm and decoupled from biological understanding. New terminology similar to taxonomy is required to facilitate the study of cell types, both common and rare. In this framework, a

cell type would be equivalent to the concept of a "species". While a specific definition of a species exists (a set of all organisms such that any two individuals of the appropriate mating types can produce fertile offspring) there is no such definition for a cell type. The more cells are captured by an experiment the more fine-grained the reported "cell types" become, similar to dividing *Homo sapiens* population into nationalities (based on their spacial location) and professions (based on their current activity). The work by Zimmermann et al. [2019] is an advancement in the process of developing useful theoretical concepts for this new field. They focused on remarkable conservation of synteny amongst ancient metazoan genes and identified genomic regions conferring ancient cell type identities. Kotliar et al. [2019] advocated for a view that considers a cell as a combination of a single cell identity (for example, being a melanocyte) and any number of cell activity programs (for example, undergoing cell division and/or responding to shortage of nutrients). They used matrix factorisation to convert a scRNA-seq expression matrix to two lower rank matrices - one encoding gene expression programs and another encoding the relative levels of activity of these programs in each cell. In mathematical terms cell types can be defined as stable attractive states, where differentiation to a different cell type (transitions) correspond to a path between them (possibly through unstable attractive states) and temporary activities (stress response, undergoing cell division, etc.) correspond to perturbations followed by return to the original stable attractive state.

The field of single cell level studies is characterised by a rapid improvement of

experimental techniques. For example, to address the noise in scRNA-seq data associated with the stochastic nature of RNA reversetranscription to cDNA Garalde et al. [2018] proposed nanopore direct RNA-seq - a protocol that produces full-length, strand-specific RNA sequences bypassing the reversetranscription step. Verboom et al. [2019] introduced a single cell strand-specific total RNA library preparation method called SMARTer. The main advantage of this novel method is that apart from polyadenylated RNA it also quantifies the transcription level of non-polyadenylated genes and circular RNAs. Additionally, SMARTer method provides the strand information of the captured transcripts. Verboom et al. [2019] demonstrated that this protocol results in a higher number of detected RNAs compared to classic scRNA-seq. [Ramani et al., 2020] developed single-cell combinatorial indexed chromosome conformation capture (sci-Hi-C) protocol - a high throughput method that produces chromatin interactome mappings in large number of single cells. This addresses the limitation of the bulk Hi-C assays that allowed for mapping of three-dimensional genome organization but were unable to account for heterogeneity of chromosome higher order structures among individual cells. To characterize spatial gene expression patterns at single cell resolution Rodriques et al. [2019] developed Slide-seq. This protocol involves transferring RNA from tissue sections onto a surface covered in DNA-barcoded beads with known positions, thus allowing the locations of the RNA to be inferred by sequencing. The resulting data allows to localize the position of the cell types previously identified in scRNA-seq datasets in a particular tissue, as well as to infer the temporal evolution of cell typespecific responses. See Stark et al. [2019] for a comprehensive

review of developments in RNA-seq, including new/improved methods available for studying RNA localisation, structure and translation, as well as long-read and direct RNA-seq technologies. These developments are likely to be incorporated in single cell protocols soon.

Undoubtedly the most exciting development in the single cell field is the new range of methods for simultaneous profiling of multiple types of data within a single cell. These techniques allow to build a much more comprehensive molecular view of the cell and thus lead to an increased understanding of the context in which transcriptomic data should be interpreted. See Stuart and Satija [2019] for a review of recent advances in integration of single cell gene expression data with other types of single cell measurements, including epigenetic, spatial, proteomic and lineage information. Since the publication of this review, new protocols have been presented. seqFISH+ developed by Eng et al. [2019] captures transcriptomic and spatial localisation data (by imaging mRNAs for up to ten thousand genes) as well as ligandreceptor pairs across neighbouring cells. scDam&T-seq proposed by Rooijers et al. [2019] allows to simultaneously quantify protein-DNA contacts by combining single-cell DNA adenine methyltransferase identification (DamID) with mRNA sequencing of the same cell. This protocol enables the analysis of protein-DNA binding related mechanisms that regulate cell type-specific transcriptional programs in heterogeneous tissues. Cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq) introduced by Stoeckius et al. [2017] uses oligonucleotide-labelled antibodies to integrate cellular protein and transcriptome

measurements into a single readout from an individual cell. The main limitation of this method is the high background from non-specific binding of antibodies which results in low resolution. Lin et al. [2019a] proposed a method that also simultaneously measures the amount of mRNA and proteins in a single cells, but unlike CITE-seq is compatible with intracellular proteins and has higher resolution. Cao et al. [2018] proposed sci-CAR - a combinatorial indexingbased co-assay that simultaneously measures chromatin accessibility and mRNA (CAR) in single cells. This method was followed by single-cell chromatin accessibility and transcriptome sequencing (scCAT-seq) protocol introduced by Liu et al. [2019]. scMethyl-HiC developped by Li et al. [2019a] simultaneously captures the chromatin conformation and DNA methylome in single cells. It is likely that this protocol will be further improved to allow for additional transcriptomic data capture. [Kimmey et al., 2019] introduced Simultaneous Overview of tri-Molecule Biosynthesis (SOM3B) - a method that simultaneously quantifies DNA, RNA, and protein synthesis in individual cells. The captures information on how DNA, RNA, and protein synthesis activities are coordinated during transient and sparse cellular processes allows to study processes such as cell differentiation. While all of the above methods provide transcriptomic data and an additional modality or two, [Mimitou et al., 2019] proposed CRISPR-compatible cellular indexing of transcriptomes and epitopes by sequencing (ECCITE-seq) for the high-throughput characterization of at least five modalities of information from each single cell.

Apart from methods that simultaneous profile multiple types of data within single

cells, another exciting direction is defined by experimental techniques that are moving beyond a snapshot of a cell. scRNA-seq approaches measuring stable RNAs provide only a snapshot of gene expression and convey little information on the true temporal dynamics and stochastic nature of transcription. Quantification of nascent RNAs is required to study immediate regulatory changes in a cell in response to internal or external clues. Manno et al. [2018] introduced the concept of RNA velocity - the time derivative of the gene expression state. They proposed to estimate RNA velocity by distinguishing between unspliced and spliced mRNAs in the transcriptomic data. Erhard et al. [2019] proposed single-cell, thiol-(SH)-linked alkylation of RNA for metabolic labelling sequencing (scSLAM-seq) which differentiates between new and old RNAs by integrating scRNA-seq with metabolic RNA labelling and biochemical nucleotide conversion. This was followed by new transcriptome alkylation-dependent single-cell RNA sequencing (NASC-seq) developed by Hendriks et al. [2019]. Identification of newly synthesised and pre-existing transcripts in single cells by NASC-seq relies on the incorporation of 4-thiouridine (4sU) into newly synthesised RNA during gene transcription and subsequent biochemical separation of 4sU-labelled and unlabelled RNA. See Wissink et al. [2019] for a comprehensive critical evaluation of the methods that add temporal dimension to single cell transcriptomic data.

5.4 Future of machine learning in biology

To make the most of the opportunities presented by the new experimental techniques described above, the analysis methods evolving alongside them are required. Current analysis methods for single cell data can be broadly divided into two categories - statistical methods and machine learning based methods (including GNNs). An opinion piece by Olhede and Wolfe [2018] considers the future of statistical methods - will they develop to meet the needs of the new types of data or be permanently replaced by machine learning. In Section 5.1 I've discussed how the flexibility of GNN architecture is the main advantage of these types of methods. In addition, GNN-based methods are scalable and make less (untestible) assumptions about the data. For example, regulatory relationships between the genes are not assumed to be constant in all cells in the dataset. A new frontier for machine learning based methods is to be able to analyse single cell data that simultaneously captures several modalities. It is likely that both supervised and unsupervised methods will be useful, while it is tempting to speculate that the major advancements will come from semi-supervised methods. Currently most progress in machine learning is made in applications to image-based problems, which leads to such nonsense as visualizing DNA sequence alignments in image form (using colours to represent both the nucleotide and the quality of the reads that are associated with it) to train a neural network that would be able to identify single nucleotide polymorphisms (SNPs) and deletions [Cai et al., 2019]. Theoretical advancements in applications of machine learning to panel data are urgently required.

There have been numerous recent advancements in GNN research that can lead to progress in GNN applications to scRNA-seq data. One promising direction is data representation in latent space. Some representations can entangle or hide features present in the data while others can facilitate learning, consequently the success of GNN-based methods generally depends on data representation. Current GNN-based methods for scRNA-seq data analysis produce a lower dimensional embedding of the data into Euclidean space, which might not be a suitable approach. Several alternatives have been proposed - uniform distribution on d-dimensional torus Mikulski and Duda [2019], hyperbolic spaces with negative curvature [Mathieu et al., 2019], etc. An interesting approach was suggested by [Muscoloni and Cannistraci, 2019] - a hyperbolic space where radial coordinate of the data points characterize their hierarchy and the angular distance between them represent their similarity. Apart from useful data representation, finding a way to disentangle the features in latent dimensions would lead to a major breakthrough in GNN applications to scRNA-seq data. In Section 4.3.1 I described my work aimed at creating disentangled latent space where varying one dimension in the latent space while other dimensions remain fixed would result exclusively in the variation of the aspect associated with this particular latent dimension. Recently a new methods for producing disentangled latent vectors have been proposed by Kim et al. [2019]. Whether or not a Euclidean space is used for data representation in latent space, there are interesting opportunities to explore non-linear interpolation in latent space, provided that the roadblocks of unequal data density have been avoided. White [2016] suggested a way to replace linear interpolation in latent space with spherical linear interpola-

tion. For a comprehensive review of recent work in the area of manifold learning, data representation in latent space and interpolation in latent space see Bengio et al. [2012].

Other aspects of GNN architecture that should be considered for improving the performance of GNNs on scRNA-seq data are the choise of learning rate, the loss function used and the additional information provided as an input. Fort et al. [2019] showed that small learning rates lead to initial learning of more specific features that prevents subsequent efficient learning from the whole dataset. This suggests that strategies that initiate the GNN training with a high learning rate (i.e. high speed of updating the parameters of the network) and later switch to a smaller learning rate could be successful. The mean square error is the most commonly used loss function that is often used in GNN applications to scRNA-seq data, including my work. One way to improve on this would be to find a more suitable loss function, for example compare the expression profile and the decoded expression profile as two discrete distributions (without log-transforming the data). Cross-entropy loss function can be combined either simply with sigmoid activation function in the output layer, or with softmax activation function in the output layer and expression profiles that are scaled to sum up to 1. Another way to improve on this would be to add additional terms to the loss function, for example a term that rewards reconstruction of expression values with similar relationships between pairs of genes. Adding *a priory* known biological information, such as a gene interaction graph, to a GNN can ensure that the train GNN will be biologically interpretable. Dutil

et al. [2018] proposed a way how a gene interaction graphs can be used to impose a bias similar to the spatial bias imposed by convolutions on an image, with a caveat that presence of many irrelevant genes in the data hinders the performance. To improve the performance of GNNs to cell type discovery in scRNA-seq data Kundu et al. [2019] proposed GAN-Tree that uses a mode-splitting algorithm to split the parent mode into semantically cohesive children modes thus facilitating unsupervised clustering. The advantages of this method include the fact that there is no assumption about the number of cell types present in the data and that new data with new cell types can be added to a trained GAN-Tree by updating only a single branch of the tree structure. In terms of using the latent space to calculate useful distances between the cells in a dataset, the work by Balcan et al. [2019] exploring data-driven algorithm selection and metric learning for clustering problem is highly relevant. They proposed strategies for simultaneously learning the best algorithm and metric for a specific clustering problem application. This exploits the fact that there are multiple ways to measure distances between data points and that the best clustering performance might require a non-trivial combination of those metrics.

Accumulation of good quality scRNA-seq data will lead to unprecedented opportunities in combining data from different experiments and different labs to accelerate scientific progress beyond what is possible in a single lab. The opportunities are both in combining scRNA-seq data sets and in combining scRNA-seq data with less noisy bulk RNA-seq data. Butler et al. [2018] introduced a strategy for integrating scRNA-seq data sets (including data sets from different species and data

sets generated via different experimental protocols) based on common sources of variation, thus enabling the identification of shared cell types across data sets. This was followed by scMerge proposed by Lin et al. [2019b] - a method for scRNA-seq data set integration based on the knowledge of genes that appear not to change across all samples. Going one step further, Wang et al. [2019a] proposed SAVER-X - a GNN-based method that uses transfer learning to not only integrate the data sets but also improve their quality by using the information available in other data sets. Ma and Pellegrini [2019] developed ACTINN - a method to train a GNN on a dataset with known cell types and subsequently use this trained GNN to assign cell types to new data. [de Kanter et al., 2019] proposed a cell type identification algorithm CHETAH that first produces a classification tree inferred from an annotated scRNA-seq dataset ant then rapidly assigns cell types to cells in a new data set. One advantage of this method is that it provides a confidence score based on the variance in gene expression per cell type. Another advantage is that it is able to predict intermediate/unassigned cell types thus preventing misclassification and providing information for previously unexplored tissues and malignant cells. Monumental efforts like The Human Cell Atlas [Regev et al., 2017] are aimed at creating a large pool of data that can be used for large scale studies with large potential for biological discovery. Mereu et al. [2019] suggested that detailed guidelines and standards should be provided to the labs wishing to contribute data to such large scale collaborative efforts. This is motivated by their study comparing 13 commonly used scRNA-seq protocols which revealed stark differences in protocol performance. The work by Svensson and da Veiga Beltrame [2019] is the first effort

to create and maintain a database of scRNA-seq studies with descriptions of what kind of data (the technique used, the number of cells profiled) and what biological systems (organism, developmental stage, tissue) have been studied, as well as the location of the data. Instead of relying on a single group of people to maintain such an essential database, it would be great to see a crowd-sourcing effort where each group when publishing a scRNA-seq study would also add it to the global repository listing all scRNA-seq studies published so far.

Machine learning based approaches have been shown to be useful for scRNA-seq data analysis, but there is no conclusive evidence that they perform better than Bayesian models or even PCA on all data sets. The new frontier where machine learning techniques will undoubtedly take the lead is the integration of scRNA-seq data with measurements of genome variants, epigenomes, proteomes, spacial transcript organisation and chromatin organization. Argelaguet et al. [2018] proposed Multi-Omics Factor Analysis (MOFA) - a computational method for discovering the principal sources of variation in multi-omics data sets. This unsupervised integration method infers a set of hidden factors that capture biological and technical sources of variability and disentangles axes of heterogeneity that are shared across multiple modalities and those specific to individual data modalities. [Stuart et al., 2019] developed a strategy to integrate different modalities of single cell level data by "anchoring" them with pairs of corresponding cells between data sets. They demonstrated the performance of their method by integrating scRNA-seq data with scATAC-seq data. See Colomé-Tatché and Theis [2018] and Stuart and Satija

[2019] for a review of existing computational methods for the analysis and integration of different -omics data types and a discussion about what new approaches are needed to leverage the full potential of single cell multi-omics data. Apart from different modalities of single cell level data, methods capable of integrating different types of evidence are also required. An approach to analyse scRNA-seq data in combination with phenotypic measurements introduced by [Saint et al., 2019] is an example of early developments in this direction. Wang et al. [2019b] developed a tool for predicting cellcell communication networks that is capable of reconstruction of complex cell lineages, including feedback or feed-forward interactions. It uses a structured cell-to-cell similarity matrix to perform unsupervised clustering, pseudotemporal ordering, lineage inference, and marker gene identification. Klimm et al. [2019] proposed a method for integrating scRNA-seq data with proteinprotein interaction networks to detect active modules in cells of different transcriptional states.

Given this wealth of opportunities for developing new computational methods to facilitate scientific discovery, an avalanche of new methods is expected. This in turn creates a pressing need for methods to generate synthetic scRNA-seq data that could be used for benchmarking these methods. The main difficulty with benchmarking methods for single cell data analysis is the lack of ground truth. For GNN-based methods, in contrast to GNNs trained on images or text, validating and visualising their performance is hindered by our inability to distinguish between a plausibly looking expression profile and an implausible one. A method to generate synthetic

scRNA-seq data would allow to assess the performance of the methods during their development, instead of the current situation of streams of methods being published and subsequently compared in review papers but almost never adopted for data analysis outside of the lab where a method was created. Currently, Splatter [Zappia et al., 2017] is the most popular method for generating synthetic scRNA-seq data. It works by fitting distributional models based on a mixture of Gamma and Poisson distributions to observed data and then drawing from these distributions. It tries to account for both biological variation and variation in library sizes, as well as technical dropouts and outlier values present in real data. Zhang et al. [2018] proposed a method that accounts not only for the expression variability across different cell types but also for the frequency of transcription bursts (that adds variability to an otherwise homogenous population of cells) and technical variation (mRNA capture, reverse transcription, RNA fragmentation, and sequencing all bundled up in a single parameter). Dibaeinia and Sinha [2019] introduced SERGIO - a method that simulates scRNA-seq data by modelling the stochastic nature of transcription as well as linear and non-linear influences of multiple transcription factors on genes according to a user-provided gene regulatory network. The advantages of SERGIO include its ability to simulate branching cell differentiation trajectories according to a user provided trajectory, and its ability to generate both unspliced and spliced transcript counts. SPsimSeq proposed by Assefa et al. [2019] generates synthetic scRNA-seq data by first constructing an empirical distribution of real gene expression data provided as input. While these methods provide a range of options enabling the user to choose how much data to provide (from as little as just a number of genes

and a number of cell types to as much as gene regulatory networks, frequency of transcription bursts and example expression data), there is a major problem with all of these methods - they do not account for the fact that scRNA-seq data is compositional data. While peer-reviewed paper addressing this need are still lacking, McGee et al. [2019] proposed the first method to generate synthetic scRNA-seq data while accounting for its compositional nature.

5.5 Conclusion

In this chapter I summarised the contributions of my work and provided my perspective on the future of single cell techniques and the future of machine learning applications in this field. My conclusion is that the main advantage of using GNN-based approaches for single cell data analysis is the inherent flexibility of these methods. In terms of a simple feed forward autoencoder, the future improvements are likely to come from development of more suitable loss functions and identifying sequences of learning rates required for different stages of training process. An important direction for future work is developing an approach for partitioning the features present in the data across the nodes of the latent layer in a way that facilitates correspondence between the nodes and the biologically meaningful features. To allow the use of GNNs for independent biological discovery (unaided by other methods or additional lab experiments) and clinical applications extensive work addressing reproducibility of GNN training is required. Major breakthroughs are likely to come from research focused on developing biologically inspired au-

toencoder architectures that are more in-line with our understanding of information flow within and between the cells than a simple feed forward autoencoder. Machine learning based approaches are likely to result in the most useful methods for analysing single cell data capturing several modalities and for integrating data sets from different experiments and labs.

Bibliography

F. Abascal, D. Juan, I. Jungreis, L. Martinez, M. Rigau, J. M. Rodriguez, J. Vazquez, and M. L. Tress. Loose ends: almost one in five human genes still have unresolved coding status. *Nucleic Acids Research*, 46(14):7070–7084, June 2018. doi: 10.1093/nar/gky587. URL https://doi.org/10.1093/nar/gky587.

F. Abdolhosseini, B. Azarkhalili, A. Maazallahi, A. Kamal, S. A. Motahari, A. Sharifi-Zarchi, and H. Chitsaz. Cell identity codes: Understanding cell identity from gene expression profiles using deep neural networks. *Scientific Reports*, 9(1), Feb. 2019. doi: 10.1038/s41598-019-38798-y. URL https://doi.org/10.1038/s41598-019-38798-y.

S. Aibar, C. B. González-Blas, T. Moerman, V. A. Huynh-Thu, H. Imrichova, G. Hulselmans, F. Rambow, J.-C. Marine, P. Geurts, J. Aerts, J. van den Oord, Z. K. Atak, J. Wouters, and S. Aerts. SCENIC: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11):1083–1086, Oct. 2017. doi: 10.1038/nmeth.4463. URL https://doi.org/10.1038/nmeth.4463.

J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal*

Statistical Society. Series B (Methodological), 44(2):139–177, 1982. ISSN 00359246. URL http://www.jstor.org/stable/2345821.

N. Aizarani, A. Saviano, Sagar, L. Mailly, S. Durand, J. S. Herman, P. Pessaux, T. F. Baumert, and D. Grün. A human liver cell atlas reveals heterogeneity and epithelial progenitors. *Nature*, 572(7768):199–204, July 2019. doi: 10.1038/s41586-019-1373-2. URL https://doi.org/10.1038/s41586-019-1373-2.

C. B. Albertin, O. Simakov, T. Mitros, Z. Y. Wang, J. R. Pungor, E. Edsinger-Gonzales, S. Brenner, C. W. Ragsdale, and D. S. Rokhsar. The octopus genome and the evolution of cephalopod neural and morphological novelties. *Nature*, 524 (7564):220–224, Aug. 2015. doi: 10.1038/nature14668. URL https://doi.org/10.1038/nature14668.

A. M. Altenhoff, N. M. Glover, C.-M. Train, K. Kaleb, A. W. Vesztrocy, D. Dylus, T. M. de Farias, K. Zile, C. Stevenson, J. Long, H. Redestig, G. H. Gonnet, and C. Dessimoz. The OMA orthology database in 2018: retrieving evolutionary relationships among all domains of life through richer web and programmatic interfaces. *Nucleic Acids Research*, 46(D1):D477–D485, Nov. 2017. doi: 10.1093/nar/gkx1019. URL https://doi.org/10.1093/nar/gkx1019.

M. Amodio, D. van Dijk, K. Srinivasan, W. S. Chen, H. Mohsen, K. R. Moon, A. Campbell, Y. Zhao, X. Wang, M. Venkataswamy, A. Desai, V. Ravi, P. Kumar, R. Montgomery, G. Wolf, and S. Krishnaswamy. Exploring single-cell data with

deep multitasking neural networks. *bioRxiv*, Dec. 2017. doi: 10.1101/237065. URL https://doi.org/10.1101/237065.

L. Amrhein, K. Harsha, and C. Fuchs. A mechanistic model for the negative binomial distribution of single-cell mRNA counts. *bioRxiv*, June 2019. doi: 10.1101/657619. URL https://doi.org/10.1101/657619.

S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10), Oct. 2010. doi: 10.1186/gb-2010-11-10-r106. URL https://doi.org/10.1186/gb-2010-11-10-r106.

T. S. Andrews and M. Hemberg. False signals induced by single-cell imputation. *F1000Research*, 7:1740, Mar. 2019. doi: 10.12688/f1000research.16613.2. URL https://doi.org/10.12688/f1000research.16613.2.

A. F. Ángyán, A. Perczel, and Z. Gáspári. Estimating intrinsic structural preferences of de novo emerging random-sequence proteins: Is aggregation the main bottleneck? *FEBS Letters*, 586(16):2468–2472, June 2012. doi: 10.1016/j.febslet.2012.06.007. URL https://doi.org/10.1016/j.febslet.2012.06.007.

Z. Arendsee, J. Li, U. Singh, P. Bhandary, A. Seetharam, and E. S. Wurtele. fagin: synteny-based phylostratigraphy and finer classification of young genes. *BMC Bioinformatics*, 20(1):440, Aug. 2019. doi: 10.1186/s12859-019-3023-y. URL https://doi.org/10.1186/s12859-019-3023-y.

R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buet-

tner, W. Huber, and O. Stegle. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6), June 2018. doi: 10.15252/msb.20178124. URL https://doi.org/10.15252/msb.20178124.

C. Arisdakessian, O. Poirion, B. Yunits, X. Zhu, and L. X. Garmire. DeepImpute: an accurate, fast and scalable deep neural network method to impute single-cell RNA-seq data. *bioRxiv*, June 2018. doi: 10.1101/353607. URL https://doi.org/10.1101/353607.

D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 233–242. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305406.

A. T. Assefa, J. Vandesompele, and O. Thas. SPsimSeq: semi-parametric simulation of bulk and single cell RNA sequencing data. *bioRxiv*, June 2019. doi: 10.1101/677740. URL https://doi.org/10.1101/677740.

R. Athanasiadou, B. Neymotin, N. Brandt, W. Wang, L. Christiaen, D. Gresham, and D. Tranchina. A complete statistical model for calibration of RNA-seq counts using external spike-ins and maximum likelihood theory. *PLOS Computational Biology*, 15(3):e1006794, Mar. 2019. doi: 10.1371/journal.pcbi.1006794. URL https://doi.org/10.1371/journal.pcbi.1006794.

E. Azizi, S. Prabhakaran, A. Carr, and D. Pe'er. Bayesian inference for single-cell clustering and imputing. *Genomics and Computational Biology*, 3(1):46, Jan. 2017. doi: 10.18547/gcb.2017.vol3.iss1.e46. URL https://doi.org/10.18547/gcb.2017.vol3.iss1.e46.

H. T. Baalsrud, O. K. Tørresen, M. H. Solbakken, W. Salzburger, R. Hanel, K. S. Jakobsen, and S. Jentoft. De novo gene evolution of antifreeze glycoproteins in codfishes revealed by whole genome sequence data. *Molecular Biology and Evolution*, 35(3):593–606, Dec. 2017. doi: 10.1093/molbev/msx311. URL https://doi.org/10.1093/molbev/msx311.

R. Bacher and C. Kendziorski. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology*, 17(1), Apr. 2016. doi: 10.1186/s13059-016-0927-y. URL https://doi.org/10.1186/s13059-016-0927-y.

R. Bacher, L.-F. Chu, N. Leng, A. P. Gasch, J. A. Thomson, R. M. Stewart, M. Newton, and C. Kendziorski. SCnorm: robust normalization of single-cell RNA-seq data. *Nature Methods*, 14(6):584–586, Apr. 2017. doi: 10.1038/nmeth.4263. URL https://doi.org/10.1038/nmeth.4263.

E. S. Balakirev and F. J. Ayala. Pseudogenes: Are they "junk" or functional DNA? *Annual Review of Genetics*, 37(1):123–151, Dec. 2003. doi: 10.1146/annurev.genet.37.040103.103949. URL https://doi.org/10.1146/annurev.genet.37.040103.103949.

M. Balcan, T. Dick, and M. Lang. Learning to link. *arXiv*, abs/1907.00533, 2019. URL http://arxiv.org/abs/1907.00533.

M. Barsacchi, H. A. Terre, and P. Lió. GEESE: Metabolically driven latent space learning for gene expression data. *bioRxiv*, July 2018. doi: 10.1101/365643. URL https://doi.org/10.1101/365643.

C. R. Bartman, S. C. Hsu, C. C.-S. Hsiung, A. Raj, and G. A. Blobel. Enhancer regulation of transcriptional bursting parameters revealed by forced chromatin looping. *Molecular Cell*, 62(2):237–247, Apr. 2016. doi: 10.1016/j.molcel.2016.03.007. URL https://doi.org/10.1016/j.molcel.2016.03.007.

A. Baser, M. Skabkin, S. Kleber, Y. Dang, G. S. G. Balta, G. Kalamakis, M. Göpferich, D. C. Ibañez, R. Schefzik, A. S. Lopez, E. L. Bobadilla, C. Schultz, B. Fischer, and A. Martin-Villalba. Onset of differentiation is post-transcriptionally controlled in adult neural stem cells. *Nature*, 566(7742):100–104, Jan. 2019. doi: 10.1038/s41586-019-0888-x. URL https://doi.org/10.1038/s41586-019-0888-x.

W. Basile, O. Sachenkova, S. Light, and A. Elofsson. High GC content causes orphan proteins to be intrinsically disordered. *PLOS Computational Biology*, 13 (3):e1005375, Mar. 2017. doi: 10.1371/journal.pcbi.1005375. URL https://doi.org/10.1371/journal.pcbi.1005375.

D. J. Begun, H. A. Lindfors, M. E. Thompson, and A. K. Holloway. Recently evolved genes identified from drosophila yakuba and d. erecta acces-

sory gland expressed sequence tags. *Genetics*, 172(3):1675–1681, Dec. 2005. doi: 10.1534/genetics.105.050336. URL https://doi.org/10.1534/genetics.105.050336.

N. Beliakova-Bethell, M. Massanella, C. White, S. M. Lada, P. Du, F. Vaida, J. Blanco, C. A. Spina, and C. H. Woelk. The effect of cell subset isolation method on gene expression in leukocytes. *Cytometry Part A*, 85(1):94–104, Sept. 2013. doi: 10.1002/cyto.a.22352. URL https://doi.org/10.1002/cyto.a.22352.

S. C. Bendall, K. L. Davis, E. ad David Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe'er. Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell*, 157(3):714–725, Apr. 2014. doi: 10.1016/j.cell.2014.04.005. URL https://doi.org/10.1016/j.cell.2014.04.005.

Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *arXiv*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin Heidelberg, 1999. doi: 10.1007/3-540-49257-7_15. URL https://doi.org/10.1007/3-540-49257-7_15.

W. Blevins, M. Albà, and L. Carey. Comparative transcriptomics and ribo-seq:

Looking at de novo gene emergence in saccharomycotina. *PeerJ*, page 3030, June 2017. doi: 10.7287/peerj.preprints.3030. URL https://doi.org/10.7287/peerj.preprints.3030.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

J. Brosius. Waste not, want not – transcript excess in multicellular eukaryotes. *Trends in Genetics*, 21(5):287–288, May 2005. doi: 10.1016/j.tig.2005.02.014. URL https://doi.org/10.1016/j.tig.2005.02.014.

F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, Jan. 2015. doi: 10.1038/nbt.3102. URL https://doi.org/10.1038/nbt.3102.

D. Bungard, J. S. Copple, J. Yan, J. J. Chhun, V. K. Kumirov, S. G. Foy, J. Masel, V. H. Wysocki, and M. H. Cordes. Foldability of a natural de novo evolved protein. *Structure*, 25(11):1687–1696.e4, Nov. 2017. doi: 10.1016/j.str.2017.09.006. URL https://doi.org/10.1016/j.str.2017.09.006.

D. B. Burkhardt, J. S. Stanley, A. L. Perdigoto, S. A. Gigante, K. C. Herold, G. Wolf, A. Giraldez, D. van Dijk, and S. Krishnaswamy. Enhancing experimental signals

in single-cell RNA-sequencing data using graph signal processing. *bioRxiv*, Jan. 2019. doi: 10.1101/532846. URL https://doi.org/10.1101/532846.

A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420, Apr. 2018. doi: 10.1038/nbt.4096. URL https://doi.org/10.1038/nbt.4096.

M. Büttner, Z. Miao, F. A. Wolf, S. A. Teichmann, and F. J. Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nature Methods*, 16(1):43–49, Dec. 2018. doi: 10.1038/s41592-018-0254-1. URL https://doi.org/10.1038/s41592-018-0254-1.

J. Cai, R. Zhao, H. Jiang, and W. Wang. De novo origination of a new protein-coding gene in saccharomyces cerevisiae. *Genetics*, 179(1):487–496, May 2008. doi: 10.1534/genetics.107.084491. URL https://doi.org/10.1534/genetics.107.084491.

L. Cai, Y. Wu, and J. Gao. DeepSV: Accurate calling of genomic deletions from high throughput sequencing data using deep convolutional neural network. *bioRxiv*, Feb. 2019. doi: 10.1101/561357. URL https://doi.org/10.1101/561357.

C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10

(1):421, 2009. doi: 10.1186/1471-2105-10-421. URL https://doi.org/10.1186/1471-2105-10-421.

P. G. Camara. Methods and challenges in the analysis of single-cell RNA-sequencing data. *Current Opinion in Systems Biology*, 7:47–53, Feb. 2018. doi: 10.1016/j.coisb.2017.12.007. URL https://doi.org/10.1016/j.coisb.2017.12.007.

E. Cannavò, N. Koelling, D. Harnett, D. Garfield, F. P. Casale, L. Ciglar, H. E. Gustafson, R. R. Viales, R. Marco-Ferreres, J. F. Degner, B. Zhao, O. Stegle, E. Birney, and E. E. M. Furlong. Genetic variants regulating expression levels and isoform diversity during embryogenesis. *Nature*, 541(7637):402–406, Dec. 2016. doi: 10.1038/nature20802. URL https://doi.org/10.1038/nature20802.

R. Cannoodt, W. Saelens, and Y. Saeys. Computational methods for trajectory inference from single-cell transcriptomics. *European Journal of Immunology*, 46(11):2496–2506, Oct. 2016. doi: 10.1002/eji.201646347. URL https://doi.org/10.1002/eji.201646347.

C. Cao, L. A. Lemaire, W. Wang, P. H. Yoon, Y. A. Choi, L. R. Parsons, J. C. Matese, W. Wang, M. Levine, and K. Chen. Comprehensive single-cell transcriptome lineages of a proto-vertebrate. *Nature*, 571(7765):349–354, July 2019a. doi: 10.1038/s41586-019-1385-y. URL https://doi.org/10.1038/s41586-019-1385-y.

J. Cao, D. A. Cusanovich, V. Ramani, D. Aghamirzaie, H. A. Pliner, A. J. Hill, R. M. Daza, J. L. McFaline-Figueroa, J. S. Packer, L. Christiansen, F. J. Steemers, A. C. Adey, C. Trapnell, and J. Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, Aug. 2018. doi: 10.1126/science.aau0730. URL https://doi.org/10.1126/science.aau0730.

J. Cao, M. Spielmann, X. Qiu, X. Huang, D. M. Ibrahim, A. J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F. J. Steemers, C. Trapnell, and J. Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566 (7745):496–502, Feb. 2019b. doi: 10.1038/s41586-019-0969-x. URL https://doi.org/10.1038/s41586-019-0969-x.

A.-R. Carvunis, T. Rolland, I. Wapinski, M. A. Calderwood, M. A. Yildirim, N. Simonis, B. Charloteaux, C. A. Hidalgo, J. Barbette, B. Santhanam, G. A. Brar, J. S. Weissman, A. Regev, N. Thierry-Mieg, M. E. Cusick, and M. Vidal. Proto-genes and de novo gene birth. *Nature*, 487(7407):370–374, June 2012. doi: 10.1038/nature11184. URL https://doi.org/10.1038/nature11184.

G. Casari, A. de Daruvar, C. Sander, and R. Schneider. Bioinformatics and the discovery of gene function. *Trends in Genetics*, 12(7):244–245, July 1996. doi: 10.1016/0168-9525(96)30057-7. URL https://doi.org/10.1016/0168-9525(96)30057-7.

C. Casola. From de novo to 'de nono': The majority of novel protein coding genes identified with phylostratigraphy are old genes or recent duplicates. *Genome*

Biology and Evolution, 10(11):2906–2918, Oct. 2018. doi: 10.1093/gbe/evy231. URL https://doi.org/10.1093/gbe/evy231.

M. Catala and S. A. Elela. Promoter-dependent nuclear RNA degradation ensures cell cycle-specific gene expression. *Communications Biology*, 2(1), June 2019. doi: 10.1038/s42003-019-0441-3. URL https://doi.org/10.1038/s42003-019-0441-3.

T. E. Chan, M. P. Stumpf, and A. C. Babtie. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Systems*, 5 (3):251–267.e3, Sept. 2017. doi: 10.1016/j.cels.2017.08.014. URL https://doi.org/10.1016/j.cels.2017.08.014.

A. Chao, R. L. Chazdon, R. K. Colwell, and T.-J. Shen. A new statistical approach for assessing similarity of species composition with incidence and abundance data. *Ecology Letters*, 8(2):148–159, Dec. 2004. doi: 10.1111/j.1461-0248.2004.00707.x. URL https://doi.org/10.1111/j.1461-0248.2004.00707.x.

B. Cheifet. Where is genomics going next? *Genome Biology*, 20(1), Jan. 2019. doi: 10.1186/s13059-019-1626-2. URL https://doi.org/10.1186/s13059-019-1626-2.

H.-I. H. Chen, Y.-C. Chiu, T. Zhang, S. Zhang, Y. Huang, and Y. Chen. GSAE: an autoencoder with embedded gene-set nodes for genomics functional characterization. *BMC Systems Biology*, 12(S8), Dec. 2018.

doi: 10.1186/s12918-018-0642-2. URL https://doi.org/10.1186/s12918-018-0642-2.

L. Chen, C. Cai, V. Chen, and X. Lu. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17(S1), Jan. 2016. doi: 10.1186/s12859-015-0852-1. URL https://doi.org/10.1186/s12859-015-0852-1.

M. Chen and X. Zhou. Controlling for confounding effects in single cell RNA sequencing studies using both control and target genes. *Scientific Reports*, 7(1), Oct. 2017. doi: 10.1038/s41598-017-13665-w. URL https://doi.org/10.1038/s41598-017-13665-w.

R. Chen and L. R. Varshney. Optimal recovery of missing values for non-negative matrix factorization. *bioRxiv*, May 2019. doi: 10.1101/647560. URL https://doi.org/10.1101/647560.

S. Chen and J. C. Mar. Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. *BMC Bioinformatics*, 19(1), June 2018. doi: 10.1186/s12859-018-2217-z. URL https://doi.org/10.1186/s12859-018-2217-z.

S. Chen, Y. E. Zhang, and M. Long. New genes in drosophila quickly become essential. *Science*, 330(6011):1682–1685, Dec. 2010. doi: 10.1126/science.1196380. URL https://doi.org/10.1126/science.1196380.

X. Chen, S. Jung, L. Y. Beh, S. R. Eddy, and L. F. Landweber. Combinatorial DNA

rearrangement facilitates the origin of new genes in ciliates. *Genome Biology and Evolution*, page evv172, Sept. 2015. doi: 10.1093/gbe/evv172. URL https://doi.org/10.1093/gbe/evv172.

J. B. Cheng, A. J. Sedgewick, A. I. Finnegan, P. Harirchian, J. Lee, S. Kwon, M. S. Fassett, J. Golovato, M. Gray, R. Ghadially, W. Liao, B. E. P. White, T. M. Mauro, T. Mully, E. A. Kim, H. Sbitany, I. M. Neuhaus, R. C. Grekin, S. S. Yu, J. W. Gray, E. Purdom, R. Paus, C. J. Vaske, S. C. Benz, J. S. Song, and R. J. Cho. Transcriptional programming of normal and inflamed human epidermis at single-cell resolution. *Cell Reports*, 25(4):871–883, Oct. 2018. doi: 10.1016/j.celrep.2018.09.006. URL https://doi.org/10.1016/j.celrep.2018.09.006.

K. R. Clarke, P. J. Somerfield, and M. G. Chapman. On resemblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted bray–curtis coefficient for denuded assemblages. *Journal of Experimental Marine Biology and Ecology*, 330(1):55–80, Mar. 2006. doi: 10.1016/j.jembe.2005.12.017. URL https://doi.org/10.1016/j.jembe.2005.12.017.

P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, Mar. 2009. doi: 10.1093/bioinformatics/btp163. URL https://doi.org/10.1093/bioinformatics/btp163.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Har-*

monic Analysis, 21(1):5–30, July 2006. doi: 10.1016/j.acha.2006.04.006. URL https://doi.org/10.1016/j.acha.2006.04.006.

R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, May 2005. doi: 10.1073/pnas.0500334102. URL https://doi.org/10.1073/pnas.0500334102.

M. B. Cole, D. Risso, A. Wagner, D. DeTomaso, J. Ngai, E. Purdom, S. Dudoit, and N. Yosef. Performance assessment and selection of normalization procedures for single-cell RNA-seq. *Cell Systems*, 8(4):315–328.e8, Apr. 2019. doi: 10.1016/j.cels.2019.03.010. URL https://doi.org/10.1016/j.cels.2019.03.010.

M. Colomé-Tatché and F. Theis. Statistical single cell multi-omics integration. *Current Opinion in Systems Biology*, 7:54–59, Feb. 2018. doi: 10.1016/j.coisb.2018.01.003. URL https://doi.org/10.1016/j.coisb.2018.01.003.

J. K. de Kanter, P. Lijnzaad, T. Candelli, T. Margaritis, and F. C. P. Holstege. CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic Acids Research*, June 2019. doi: 10.1093/nar/gkz543. URL https://doi.org/10.1093/nar/gkz543.

T. Y. de Soysa, S. S. Ranade, S. Okawa, S. Ravichandran, Y. Huang, H. T. Salunga, A. Schricker, A. del Sol, C. A. Gifford, and D. Srivastava. Single-cell analysis of

cardiogenesis reveals basis for organ-level developmental defects. *Nature*, 572 (7767):120–124, July 2019. doi: 10.1038/s41586-019-1414-x. URL https://doi.org/10.1038/s41586-019-1414-x.

M. Delmans and M. Hemberg. Discrete distributional differential expression (d3e) - a tool for gene expression analysis of single-cell RNA-seq data. *BMC Bioinformatics*, 17(1), Feb. 2016. doi: 10.1186/s12859-016-0944-6. URL https://doi.org/10.1186/s12859-016-0944-6.

Y. Deng, F. Bao, Q. Dai, L. F. Wu, and S. J. Altschuler. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nature Methods*, 16(4):311–314, Mar. 2019. doi: 10.1038/s41592-019-0353-7. URL https://doi.org/10.1038/s41592-019-0353-7.

P. Dibaeinia and S. Sinha. A single-cell expression simulator guided by gene regulatory networks. *bioRxiv*, July 2019. doi: 10.1101/716811. URL https://doi.org/10.1101/716811.

T. Domazet-Loso. An evolutionary analysis of orphan genes in drosophila. *Genome Research*, 13(10):2213–2219, Oct. 2003. doi: 10.1101/gr.1311003. URL https://doi.org/10.1101/gr.1311003.

Y. Dong, F. Bao, H. Su, and J. Zhu. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv*, abs/1901.09035, 2019. URL http://arxiv.org/abs/1901.09035.

M. T. Donoghue, C. Keshavaiah, S. H. Swamidatta, and C. Spillane. Evolutionary

origins of brassicaceae specific genes in arabidopsis thaliana. *BMC Evolutionary Biology*, 11(1), Feb. 2011. doi: 10.1186/1471-2148-11-47. URL https://doi.org/10.1186/1471-2148-11-47.

B. T. Donovan, A. Huynh, D. A. Ball, H. P. Patel, M. G. Poirier, D. R. Larson, M. L. Ferguson, and T. L. Lenstra. Live-cell imaging reveals the interplay between transcription factors, nucleosomes, and bursting. *The EMBO Journal*, 38(12), May 2019. doi: 10.15252/embj.2018100809. URL https://doi.org/10.15252/embj.2018100809.

Drosophila 12 Genomes Consortium. Evolution of genes and genomes on the drosophila phylogeny. *Nature*, 450(7167):203–218, Nov. 2007. doi: 10.1038/nature06341. URL https://doi.org/10.1038/nature06341.

B. Dujon. The yeast genome project: what did we learn? *Trends in Genetics*, 12 (7):263–270, July 1996. doi: 10.1016/0168-9525(96)10027-5. URL https://doi.org/10.1016/0168-9525(96)10027-5.

B. W. Dulken, M. T. Buckley, P. N. Negredo, N. Saligrama, R. Cayrol, D. S. Leeman, B. M. George, S. C. Boutet, K. Hebestreit, J. V. Pluvinage, T. Wyss-Coray, I. L. Weissman, H. Vogel, M. M. Davis, and A. Brunet. Single-cell analysis reveals t cell infiltration in old neurogenic niches. *Nature*, 571(7764):205–210, July 2019. doi: 10.1038/s41586-019-1362-5. URL https://doi.org/10.1038/s41586-019-1362-5.

F. Dutil, J. P. Cohen, M. Weiss, G. Derevyanko, and Y. Bengio. Towards gene

expression convolutions using gene interaction graphs. *arXiv*, abs/1806.06975, 2018. URL http://arxiv.org/abs/1806.06975.

S. R. Eddy. Accelerated profile HMM searches. *PLoS Computational Biology*, 7 (10):e1002195, Oct. 2011. doi: 10.1371/journal.pcbi.1002195. URL https://doi.org/10.1371/journal.pcbi.1002195.

M. A. El-Brolosy, Z. Kontarakis, A. Rossi, C. Kuenne, S. Günther, N. Fukuda, K. Kikhi, G. L. M. Boezio, C. M. Takacs, S.-L. Lai, R. Fukuda, C. Gerri, A. J. Giraldez, and D. Y. R. Stainier. Genetic compensation triggered by mutant mRNA degradation. *Nature*, 568(7751):193–197, Apr. 2019. doi: 10.1038/s41586-019-1064-z. URL https://doi.org/10.1038/s41586-019-1064-z.

C.-H. L. Eng, M. Lawson, Q. Zhu, R. Dries, N. Koulena, Y. Takei, J. Yun, C. Cronin, C. Karp, G.-C. Yuan, and L. Cai. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH +. *Nature*, 568(7751):235–239, Mar. 2019. doi: 10.1038/s41586-019-1049-y. URL https://doi.org/10.1038/s41586-019-1049-y.

J. M. Engreitz, B. J. Daigle, J. J. Marshall, and R. B. Altman. Independent component analysis: Mining microarray data for fundamental human gene expression modules. *Journal of Biomedical Informatics*, 43(6):932–944, Dec. 2010. doi: 10.1016/j.jbi.2010.07.001. URL https://doi.org/10.1016/j.jbi.2010.07.001.

G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10(1), Jan. 2019. doi: 10.1038/s41467-018-07931-2. URL https://doi.org/10.1038/s41467-018-07931-2.

I. Erb and C. Notredame. How should we measure proportionality on relative gene expression data? *Theory in Biosciences*, 135(1-2):21–36, Jan. 2016. doi: 10.1007/s12064-015-0220-8. URL https://doi.org/10.1007/s12064-015-0220-8.

F. Erhard, M. A. P. Baptista, T. Krammer, T. Hennig, M. Lange, P. Arampatzi, C. S. Jürges, F. J. Theis, A.-E. Saliba, and L. Dölken. scSLAM-seq reveals core features of transcription dynamics in single cells. *Nature*, 571(7765):419–423, July 2019. doi: 10.1038/s41586-019-1369-y. URL https://doi.org/10.1038/s41586-019-1369-y.

Y. J. Fan, J. E. Allen, S. A. Jacobs, and B. C. V. Essen. Distinguishing between normal and cancer cells using autoencoder node saliency. *arXiv*, abs/1901.11152, 2019. URL http://arxiv.org/abs/1901.11152.

K. Faust, J. F. Sathirapongsasuti, J. Izard, N. Segata, D. Gevers, J. Raes, and C. Huttenhower. Microbial co-occurrence relationships in the human microbiome. *PLoS Computational Biology*, 8(7):e1002606, July 2012. doi: 10.1371/journal.pcbi.1002606. URL https://doi.org/10.1371/journal.pcbi.1002606.

A. D. Fernandes, J. N. Reid, J. M. Macklaim, T. A. McMurrough, D. R. Edgell, and G. B. Gloor. Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16s rRNA gene sequencing and selective growth experiments by compositional data analysis. *Microbiome*, 2(1):15, 2014. doi: 10.1186/2049-2618-2-15. URL https://doi.org/10.1186/2049-2618-2-15.

A.-M. Fernandez-Escamilla, F. Rousseau, J. Schymkowitz, and L. Serrano. Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins. *Nature Biotechnology*, 22(10):1302–1306, Sept. 2004. doi: 10.1038/nbt1012. URL https://doi.org/10.1038/nbt1012.

G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. K. Shalek, C. K. Slichter, H. W. Miller, M. J. McElrath, M. Prlic, P. S. Linsley, and R. Gottardo. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(1), Dec. 2015. doi: 10.1186/s13059-015-0844-5. URL https://doi.org/10.1186/s13059-015-0844-5.

R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate, and A. Bateman. The pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 44(D1):D279–D285, Dec. 2015. doi: 10.1093/nar/gkv1344. URL https://doi.org/10.1093/nar/gkv1344.

R. Foreman and R. Wollman. Mammalian gene expression variability is explained

by underlying cell state. *bioRxiv*, May 2019. doi: 10.1101/626424. URL https://doi.org/10.1101/626424.

S. Fort, P. K. Nowak, and S. Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv*, abs/1901.09491, 2019. URL http://arxiv.org/abs/1901.09491.

S. G. Foy, B. A. Wilson, J. Bertram, M. H. J. Cordes, and J. Masel. A shift in aggregation avoidance strategy marks a long-term direction to protein evolution. *Genetics*, 211(4):1345–1355, Jan. 2019. doi: 10.1534/genetics.118.301719. URL https://doi.org/10.1534/genetics.118.301719.

J. Frigola, R. Sabarinathan, L. Mularoni, F. Muiños, A. Gonzalez-Perez, and N. López-Bigas. Reduced mutation rate in exons due to differential mismatch repair. *Nature Genetics*, 49(12):1684–1692, Nov. 2017. doi: 10.1038/ng.3991. URL https://doi.org/10.1038/ng.3991.

D. R. Garalde, E. A. Snell, D. Jachimowicz, B. Sipos, J. H. Lloyd, M. Bruce, N. Pantic, T. Admassu, P. James, A. Warland, M. Jordan, J. Ciccone, S. Serra, J. Keenan, S. Martin, L. McNeill, E. J. Wallace, L. Jayasinghe, C. Wright, J. Blasco, S. Young, D. Brocklebank, S. Juul, J. Clarke, A. J. Heron, and D. J. Turner. Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods*, 15(3):201–206, Jan. 2018. doi: 10.1038/nmeth.4577. URL https://doi.org/10.1038/nmeth.4577.

L. Garcia-Alonso, C. H. Holland, M. M. Ibrahim, D. Turei, and J. Saez-Rodriguez.

Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome Research*, 29(8):1363–1375, July 2019. doi: 10.1101/gr.240663.118. URL https://doi.org/10.1101/gr.240663.118.

R. Gaujoux and C. Seoighe. Semi-supervised nonnegative matrix factorization for gene expression deconvolution: A case study. *Infection, Genetics and Evolution*, 12(5):913–921, July 2012. doi: 10.1016/j.meegid.2011.08.014. URL https://doi.org/10.1016/j.meegid.2011.08.014.

S. Ghatak, Z. A. King, A. Sastry, and B. O. Palsson. The y-ome defines the 35% of escherichia coli genes that lack experimental evidence of function. *Nucleic Acids Research*, 47(5):2446–2454, Jan. 2019. doi: 10.1093/nar/gkz030. URL https://doi.org/10.1093/nar/gkz030.

M. P. Gold, A. LeNail, and E. Fraenkel. Shallow sparsely-connected autoencoders for gene set projection. In *Biocomputing 2019*. WORLD SCIENTIFIC, Nov. 2018. doi: 10.1142/9789813279827_0034. URL https://doi.org/10.1142/9789813279827_0034.

W. Gong, B. N. Singh, P. Shah, S. Das, J. Theisen, S. Chan, M. Kyba, M. G. Garry, D. Yannopoulos, W. Pan, and D. J. Garry. A novel algorithm for the collective integration of single cell RNA-seq during embryogenesis. *bioRxiv*, Feb. 2019. doi: 10.1101/543314. URL https://doi.org/10.1101/543314.

T. Graf and T. Enver. Forcing cells to change lineages. *Nature*, 462(7273):587–594,

Dec. 2009. doi: 10.1038/nature08533. URL https://doi.org/10.1038/nature08533.

D. Graur, Y. Zheng, N. Price, R. B. R. Azevedo, R. A. Zufall, and E. Elhaik. On the immortality of television sets: "function" in the human genome according to the evolution-free gospel of ENCODE. *Genome Biology and Evolution*, 5(3): 578–590, Feb. 2013. doi: 10.1093/gbe/evt028. URL https://doi.org/10.1093/gbe/evt028.

J. A. Griffiths, A. Scialdone, and J. C. Marioni. Using single-cell genomics to understand developmental processes and cell fate decisions. *Molecular Systems Biology*, 14(4), Apr. 2018. doi: 10.15252/msb.20178046. URL https://doi.org/10.15252/msb.20178046.

C. H. Grønbech, M. F. Vording, P. Timshel, C. K. Sønderby, T. H. Pers, and O. Winther. scVAE: Variational auto-encoders for single-cell gene expression data. *bioRxiv*, May 2018. doi: 10.1101/318295. URL https://doi.org/10.1101/318295.

D. Grün, L. Kester, and A. van Oudenaarden. Validation of noise models for single-cell transcriptomics. *Nature Methods*, 11(6):637–640, Apr. 2014. doi: 10.1038/nmeth.2930. URL https://doi.org/10.1038/nmeth.2930.

Y. Guan, L. Liu, Q. Wang, J. Zhao, P. Li, J. Hu, Z. Yang, M. P. Running, H. Sun, and J. Huang. Gene refashioning through innovative shifting of reading frames in mosses. *Nature Communications*, 9(1), Apr. 2018.

doi: 10.1038/s41467-018-04025-x. URL https://doi.org/10.1038/s41467-018-04025-x.

D. Guerzoni and A. McLysaght. De novo genes arise at a slow but steady rate along the primate lineage and have been subject to incomplete lineage sorting. *Genome Biology and Evolution*, 8(4):1222–1232, Apr. 2016. doi: 10.1093/gbe/evw074. URL https://doi.org/10.1093/gbe/evw074.

J. Guiu, E. Hannezo, S. Yui, S. Demharter, S. Ulyanchenko, M. Maimets, A. Jørgensen, S. Perlman, L. Lundvall, L. S. Mamsen, A. Larsen, R. H. Olesen, C. Y. Andersen, L. L. Thuesen, K. J. Hare, T. H. Pers, K. Khodosevich, B. D. Simons, and K. B. Jensen. Tracing the origin of adult intestinal stem cells. *Nature*, 570(7759):107–111, May 2019. doi: 10.1038/s41586-019-1212-5. URL https://doi.org/10.1038/s41586-019-1212-5.

C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *bioRxiv*, Mar. 2019. doi: 10.1101/576827. URL https://doi.org/10.1101/576827.

D. H. Haft, M. DiCuccio, A. Badretdin, V. Brover, V. Chetvernin, K. O'Neill, W. Li, F. Chitsaz, M. K. Derbyshire, N. R. Gonzales, M. Gwadz, F. Lu, G. H. Marchler, J. S. Song, N. Thanki, R. A. Yamashita, C. Zheng, F. Thibaud-Nissen, L. Y. Geer, A. Marchler-Bauer, and K. D. Pruitt. RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Research*, 46(D1):D851–D860, Nov. 2017. doi: 10.1093/nar/gkx1068. URL https://doi.org/10.1093/nar/gkx1068.

L. Haghverdi, F. Buettner, and F. J. Theis. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, 31(18):2989–2998, May 2015. doi: 10.1093/bioinformatics/btv325. URL https://doi.org/10.1093/bioinformatics/btv325.

L. Haghverdi, M. Büttner, F. A. Wolf, F. Buettner, and F. J. Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848, Aug. 2016. doi: 10.1038/nmeth.3971. URL https://doi.org/10.1038/nmeth.3971.

L. Haghverdi, A. T. L. Lun, M. D. Morgan, and J. C. Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421–427, Apr. 2018. doi: 10.1038/nbt.4091. URL https://doi.org/10.1038/nbt.4091.

T. Hashimoto, D. D. Horikawa, Y. Saito, H. Kuwahara, H. Kozuka-Hata, T. Shin-I, Y. Minakuchi, K. Ohishi, A. Motoyama, T. Aizu, A. Enomoto, K. Kondo, S. Tanaka, Y. Hara, S. Koshikawa, H. Sagara, T. Miura, S. ichi Yokobori, K. Miyagawa, Y. Suzuki, T. Kubo, M. Oyama, Y. Kohara, A. Fujiyama, K. Arakawa, T. Katayama, A. Toyoda, and T. Kunieda. Extremotolerant tardigrade genome and improved radiotolerance of human cultured cells by tardigrade-unique protein. *Nature Communications*, 7(1):12808, Sept. 2016. doi: 10.1038/ncomms12808. URL https://doi.org/10.1038/ncomms12808.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing

human-level performance on imagenet classification. *arXiv*, abs/1502.01852, 2015a. URL http://arxiv.org/abs/1502.01852.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv*, abs/1512.03385, 2015b. URL http://arxiv.org/abs/1512.03385.

G.-J. Hendriks, L. A. Jung, A. J. M. Larsson, M. Lidschreiber, O. A. Forsman, K. Lidschreiber, P. Cramer, and R. Sandberg. NASC-seq monitors RNA synthesis in single cells. *Nature Communications*, 10(1), July 2019. doi: 10.1038/s41467-019-11028-9. URL https://doi.org/10.1038/s41467-019-11028-9.

S. C. Hicks, F. W. Townes, M. Teng, and R. A. Irizarry. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 19(4):562–578, Nov. 2017. doi: 10.1093/biostatistics/kxx053. URL https://doi.org/10.1093/biostatistics/kxx053.

M. Hild, B. Beckmann, S. Haas, B. Koch, V. Solovyev, C. Busold, K. Fellenberg, M. Boutros, M. Vingron, F. Sauer, J. Hoheisel, and R. Paro. An integrated gene annotation and transcriptional profiling approach towards the full gene content of the drosophila genome. *Genome Biology*, 5(1):R3, 2003. doi: 10.1186/gb-2003-5-1-r3. URL https://doi.org/10.1186/gb-2003-5-1-r3.

R. D. Hodge, T. E. Bakken, J. A. Miller, K. A. Smith, E. R. Barkan, L. T. Gray-

buck, J. L. Close, B. Long, N. Johansen, O. Penn, Z. Yao, J. Eggermont, T. Höllt, B. P. Levi, S. I. Shehata, B. Aevermann, A. Beller, D. Bertagnolli, K. Brouner, T. Casper, C. Cobbs, R. Dalley, N. Dee, S.-L. Ding, R. G. Ellenbogen, O. Fong, E. Garren, J. Goldy, R. P. Gwinn, D. Hirschstein, C. D. Keene, M. Keshk, A. L. Ko, K. Lathia, A. Mahfouz, Z. Maltzer, M. McGraw, T. N. Nguyen, J. Nyhus, J. G. Ojemann, A. Oldre, S. Parry, S. Reynolds, C. Rimorin, N. V. Shapovalova, S. Somasundaram, A. Szafer, E. R. Thomsen, M. Tieu, G. Quon, R. H. Scheuermann, R. Yuste, S. M. Sunkin, B. Lelieveldt, D. Feng, L. Ng, A. Bernard, M. Hawrylycz, J. W. Phillips, B. Tasic, H. Zeng, A. R. Jones, C. Koch, and E. S. Lein. Conserved cell types with divergent features in human versus mouse cortex. *Nature*, 573(7772):61–68, Aug. 2019. doi: 10.1038/s41586-019-1506-7. URL `https://doi.org/10.1038/s41586-019-1506-7`.

G. Hooker and L. Mentch. Please stop permuting features: An explanation and alternatives. *arXiv*, abs/1905.03151, 2019. URL `http://arxiv.org/abs/1905.03151`.

R. Horvath, B. Laenen, S. Takuno, and T. Slotte. Single-cell expression noise and gene-body methylation in arabidopsis thaliana. *Heredity*, 123(2):81–91, Jan. 2019. doi: 10.1038/s41437-018-0181-z. URL `https://doi.org/10.1038/s41437-018-0181-z`.

M.-W. Hu, D. W. Kim, S. Liu, D. J. Zack, S. Blackshaw, and J. Qian. PanoView: An iterative clustering method for single-cell RNA sequencing data. *PLOS Computa-*

tional Biology, 15(8):e1007040, Aug. 2019a. doi: 10.1371/journal.pcbi.1007040. URL https://doi.org/10.1371/journal.pcbi.1007040.

Y. Hu, X. Wang, B. Hu, Y. Mao, Y. Chen, L. Yan, J. Yong, J. Dong, Y. Wei, W. Wang, L. Wen, J. Qiao, and F. Tang. Dissecting the transcriptome landscape of the human fetal neural retina and retinal pigment epithelium by single-cell RNA-seq analysis. *PLOS Biology*, 17(7):e3000365, July 2019b. doi: 10.1371/journal.pbio.3000365. URL https://doi.org/10.1371/journal.pbio.3000365.

M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. I. Murray, A. Raj, M. Li, and N. R. Zhang. SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15(7):539–542, June 2018. doi: 10.1038/s41592-018-0033-z. URL https://doi.org/10.1038/s41592-018-0033-z.

S. Huang, X. Xu, L. Zheng, and G. W. Wornell. An information theoretic interpretation to deep neural networks. *arXiv*, abs/1905.06600, 2019. URL http://arxiv.org/abs/1905.06600.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

T. Ilicic, J. K. Kim, A. A. Kolodziejczyk, F. O. Bagger, D. J. McCarthy, J. C. Marioni, and S. A. Teichmann. Classification of low quality cells from single-cell RNA-seq data. *Genome Biology*, 17(1), Feb. 2016.

doi: 10.1186/s13059-016-0888-1. URL https://doi.org/10.1186/s13059-016-0888-1.

S. Islam, A. Zeisel, S. Joost, G. L. Manno, P. Zajac, M. Kasper, P. Lönnerberg, and S. Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166, Dec. 2013. doi: 10.1038/nmeth.2772. URL https://doi.org/10.1038/nmeth.2772.

M. K. Jaakkola, F. Seyednasrollah, A. Mehmood, and L. L. Elo. Comparison of methods to detect differentially expressed genes between single-cell populations. *Briefings in Bioinformatics*, page bbw057, July 2016. doi: 10.1093/bib/bbw057. URL https://doi.org/10.1093/bib/bbw057.

F. Jacob. Evolution and tinkering. *Science*, 196(4295):1161–1166, June 1977. doi: 10.1126/science.860134. URL https://doi.org/10.1126/science.860134.

A. Jain, D. Perisa, F. Fliedner, A. von Haeseler, and I. Ebersberger. The evolutionary traceability of a protein. *Genome Biology and Evolution*, 11(2):531–545, Jan. 2019. doi: 10.1093/gbe/evz008. URL https://doi.org/10.1093/gbe/evz008.

N. Johansen and G. Quon. scAlign: a tool for alignment, integration, and rare cell identification from scRNA-seq data. *Genome Biology*, 20(1), Aug. 2019. doi: 10.1186/s13059-019-1766-4. URL https://doi.org/10.1186/s13059-019-1766-4.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/.

M. J. C. Jordão, R. Sankowski, S. M. Brendecke, Sagar, G. Locatelli, Y.-H. Tai, T. L. Tay, E. Schramm, S. Armbruster, N. Hagemeyer, O. Groß, D. Mai, Özgün Çiçek, T. Falk, M. Kerschensteiner, D. Grün, and M. Prinz. Single-cell profiling identifies myeloid cell subsets with distinct fates during neuroinflammation. *Science*, 363(6425):eaat7554, Jan. 2019. doi: 10.1126/science.aat7554. URL https://doi.org/10.1126/science.aat7554.

I. Jungreis, C. S. Chan, R. M. Waterhouse, G. Fields, M. F. Lin, and M. Kellis. Evolutionary dynamics of abundant stop codon readthrough. *Molecular Biology and Evolution*, 33(12):3108–3132, Sept. 2016. doi: 10.1093/molbev/msw189. URL https://doi.org/10.1093/molbev/msw189.

I. Kanter, P. Dalerba, and T. Kalisky. A cluster robustness score for identifying cell subpopulations in single cell gene expression datasets from heterogeneous tissues and tumors. *Bioinformatics*, 35(6):962–971, Aug. 2018. doi: 10.1093/bioinformatics/bty708. URL https://doi.org/10.1093/bioinformatics/bty708.

K. Katoh and D. M. Standley. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780, Jan. 2013. doi: 10.1093/molbev/mst010. URL https://doi.org/10.1093/molbev/mst010.

B. Khalfaoui and J.-P. Vert. DropLasso: A robust variant of Lasso for single cell RNA-seq data. *arXiv*, art. arXiv:1802.09381, Feb 2018.

P. V. Kharchenko, L. Silberstein, and D. T. Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, May 2014. doi: 10.1038/nmeth.2967. URL https://doi.org/10.1038/nmeth.2967.

J. K. Kim, A. A. Kolodziejczyk, T. Ilicic, S. A. Teichmann, and J. C. Marioni. Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature Communications*, 6(1), Oct. 2015. doi: 10.1038/ncomms9687. URL https://doi.org/10.1038/ncomms9687.

M. Kim, Y. Wang, P. Sahu, and V. Pavlovic. Relevance factor VAE: learning and identifying disentangled factors. *arXiv*, abs/1902.01568, 2019. URL http://arxiv.org/abs/1902.01568.

P. M. Kim. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, 13(7):1706–1718, July 2003. doi: 10.1101/gr.903503. URL https://doi.org/10.1101/gr.903503.

T. Kim, I. R. Chen, Y. Lin, A. Y.-Y. Wang, J. Y. H. Yang, and P. Yang. Impact of similarity metrics on single-cell RNA-seq data clustering. *Briefings in Bioinformatics*, Aug. 2018. doi: 10.1093/bib/bby076. URL https://doi.org/10.1093/bib/bby076.

J. C. Kimmel, L. Penland, N. D. Rubinstein, D. G. Hendrickson, D. R. Kelley, and A. Z. Rosenthal. A murine aging cell atlas reveals cell identity and tissue-specific trajectories of aging. *bioRxiv*, June 2019. doi: 10.1101/657726. URL https://doi.org/10.1101/657726.

S. C. Kimmey, L. Borges, R. Baskar, and S. C. Bendall. Parallel analysis of tri-molecular biosynthesis with cell identity and function in single cells. *Nature Communications*, 10(1), Mar. 2019. doi: 10.1038/s41467-019-09128-7. URL https://doi.org/10.1038/s41467-019-09128-7.

S. Kinalis, F. C. Nielsen, O. Winther, and F. O. Bagger. Deconvolution of autoencoders to learn biological regulatory modules from single cell mRNA sequencing data. *BMC Bioinformatics*, 20(1), July 2019. doi: 10.1186/s12859-019-2952-9. URL https://doi.org/10.1186/s12859-019-2952-9.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, abs/1412.6980, 2014.

F. Klimm, E. M. Toledo, T. Monfeuga, F. Zhang, C. M. Deane, and G. Reinert. Functional module detection through integration of single-cell RNA sequencing data with protein–protein interaction networks. *bioRxiv*, July 2019. doi: 10.1101/698647. URL https://doi.org/10.1101/698647.

M. Knopp and D. I. Andersson. No beneficial fitness effects of random peptides. *Nature Ecology & Evolution*, 2(7):1046–1047, June 2018.

doi: 10.1038/s41559-018-0585-4. URL https://doi.org/10.1038/s41559-018-0585-4.

D. G. Knowles and A. McLysaght. Recent de novo origin of human protein-coding genes. *Genome Research*, 19(10):1752–1759, Sept. 2009. doi: 10.1101/gr.095026.109. URL https://doi.org/10.1101/gr.095026.109.

E. J. Kort, M. Weiland, E. Grins, E. Eugster, H. yun Milliron, C. Kelty, N. M. Shrestha, T. Timek, M. Leacche, S. J. Fitch, T. J. Boeve, G. Marco, M. Dickinson, P. Wilton, and S. Jovinge. Single cell transcriptomics is a robust approach to defining disease biology in complex clinical settings. *bioRxiv*, Mar. 2019. doi: 10.1101/568659. URL https://doi.org/10.1101/568659.

M. J. Koster, B. Snel, and H. M. Timmers. Genesis of chromatin and transcription dynamics in the origin of species. *Cell*, 161(4):724–736, May 2015. doi: 10.1016/j.cell.2015.04.033. URL https://doi.org/10.1016/j.cell.2015.04.033.

D. Kotliar, A. Veres, M. A. Nagy, S. Tabrizi, E. Hodis, D. A. Melton, and P. C. Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-seq. *eLife*, 8, July 2019. doi: 10.7554/elife.43803. URL https://doi.org/10.7554/elife.43803.

J. N. Kundu, M. Gor, D. Agrawal, and R. V. Babu. Gan-tree: An incrementally learned hierarchical generative framework for multi-modal data distributions. *arXiv*, Sept. 2019.

S. Ladjal, A. Newson, and C.-H. Pham. A PCA-like Autoencoder. *arXiv*, art. arXiv:1904.01277, Apr 2019.

H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *J. Mach. Learn. Res.*, 10:1–40, June 2009. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1577069.1577070.

A. C. Leote, X. Wu, and A. Beyer. Network-based imputation of dropouts in single-cell RNA sequencing data. *bioRxiv*, Apr. 2019. doi: 10.1101/611517. URL https://doi.org/10.1101/611517.

M. T. Levine, C. D. Jones, A. D. Kern, H. A. Lindfors, and D. J. Begun. Novel genes derived from noncoding DNA in drosophila melanogaster are frequently x-linked and exhibit testis-biased expression. *Proceedings of the National Academy of Sciences*, 103(26):9935–9939, June 2006. doi: 10.1073/pnas.0509809103. URL https://doi.org/10.1073/pnas.0509809103.

B. Li and C. N. Dewey. RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics*, 12(1), Aug. 2011. doi: 10.1186/1471-2105-12-323. URL https://doi.org/10.1186/1471-2105-12-323.

C. Li, F. Cesbron, M. Oehler, M. Brunner, and T. Höfer. Frequency modulation of transcriptional bursting enables sensitive and rapid gene regulation. *Cell Systems*,

6(4):409–423.e11, Apr. 2018. doi: 10.1016/j.cels.2018.01.012. URL https://doi.org/10.1016/j.cels.2018.01.012.

G. Li, Y. Liu, Y. Zhang, N. Kubo, M. Yu, R. Fang, M. Kellis, and B. Ren. Joint profiling of DNA methylation and chromatin architecture in single cells. *Nature Methods*, 16(10):991–993, Aug. 2019a. doi: 10.1038/s41592-019-0502-z. URL https://doi.org/10.1038/s41592-019-0502-z.

R. Li and G. Quon. scBFA: modeling detection patterns to mitigate technical noise in large-scale single-cell genomics data. *Genome Biology*, 20(1), Sept. 2019. doi: 10.1186/s13059-019-1806-0. URL https://doi.org/10.1186/s13059-019-1806-0.

W. V. Li and J. J. Li. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature Communications*, 9(1), Mar. 2018. doi: 10.1038/s41467-018-03405-7. URL https://doi.org/10.1038/s41467-018-03405-7.

X. Li, Y. Lyu, J. Park, J. Zhang, D. Stambolian, K. Susztak, G. Hu, and M. Li. Deep learning enables accurate clustering and batch effect removal in single-cell RNA-seq analysis: Supplementary material. *bioRxiv*, Jan. 2019b. doi: 10.1101/530378. URL https://doi.org/10.1101/530378.

S. Liang, F. Wang, J. Han, and K. Chen. Latent periodic process inference from single-cell RNA-seq data. *bioRxiv*, May 2019. doi: 10.1101/625566. URL https://doi.org/10.1101/625566.

J. C. Liao, R. Boscolo, Y.-L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury. Network component analysis: Reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100 (26):15522–15527, Dec. 2003. doi: 10.1073/pnas.2136632100. URL https://doi.org/10.1073/pnas.2136632100.

J. Lin, C. Jordi, M. Son, H. V. Phan, N. Drayman, M. F. Abasiyanik, L. Vistain, H.-L. Tu, and S. Tay. Ultra-sensitive digital quantification of proteins and mRNA in single cells. *Nature Communications*, 10(1), Aug. 2019a. doi: 10.1038/s41467-019-11531-z. URL https://doi.org/10.1038/s41467-019-11531-z.

Y. Lin, S. Ghazanfar, K. Y. X. Wang, J. A. Gagnon-Bartsch, K. K. Lo, X. Su, Z.-G. Han, J. T. Ormerod, T. P. Speed, P. Yang, and J. Y. H. Yang. scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets. *Proceedings of the National Academy of Sciences*, 116(20):9775–9784, Apr. 2019b. doi: 10.1073/pnas.1820006116. URL https://doi.org/10.1073/pnas.1820006116.

L. Liu, C. Liu, A. Quintero, L. Wu, Y. Yuan, M. Wang, M. Cheng, L. Leng, L. Xu, G. Dong, R. Li, Y. Liu, X. Wei, J. Xu, X. Chen, H. Lu, D. Chen, Q. Wang, Q. Zhou, X. Lin, G. Li, S. Liu, Q. Wang, H. Wang, J. L. Fink, Z. Gao, X. Liu, Y. Hou, S. Zhu, H. Yang, Y. Ye, G. Lin, F. Chen, C. Herrmann, R. Eils, Z. Shang, and X. Xu. Deconvolution of single-cell multi-omics layers reveals regulatory heterogeneity. *Nature Communications*, 10(1), Jan. 2019.

doi: 10.1038/s41467-018-08205-7. URL https://doi.org/10.1038/s41467-018-08205-7.

M. Long, E. Betrán, K. Thornton, and W. Wang. The origin of new genes: glimpses from the young and old. *Nature Reviews Genetics*, 4(11):865–875, Nov. 2003. doi: 10.1038/nrg1204. URL https://doi.org/10.1038/nrg1204.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, Nov. 2018. doi: 10.1038/s41592-018-0229-2. URL https://doi.org/10.1038/s41592-018-0229-2.

M. Lotfollahi, F. A. Wolf, and F. J. Theis. scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, July 2019. doi: 10.1038/s41592-019-0494-8. URL https://doi.org/10.1038/s41592-019-0494-8.

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), Dec. 2014. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

D. Lovell, V. Pawlowsky-Glahn, J. J. Egozcue, S. Marguerat, and J. Bähler. Proportionality: A valid alternative to correlation for relative data. *PLOS Computational Biology*, 11(3):e1004075, Mar. 2015. doi: 10.1371/journal.pcbi.1004075. URL https://doi.org/10.1371/journal.pcbi.1004075.

M. D. Luecken and F. J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, June 2019. doi: 10.15252/msb.20188746. URL https://doi.org/10.15252/msb.20188746.

S. Lukassen, F. W. Ten, R. Eils, and C. Conrad. Gene set inference from single-cell sequencing data using a hybrid of matrix factorization and variational autoencoders. *bioRxiv*, Aug. 2019. doi: 10.1101/740415. URL https://doi.org/10.1101/740415.

A. T. L. Lun, K. Bach, and J. C. Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17 (1), Apr. 2016. doi: 10.1186/s13059-016-0947-7. URL https://doi.org/10.1186/s13059-016-0947-7.

F. Ma and M. Pellegrini. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, July 2019. doi: 10.1093/bioinformatics/btz592. URL https://doi.org/10.1093/bioinformatics/btz592.

E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, J. J. Trombetta, D. A. Weitz, J. R. Sanes, A. K. Shalek, A. Regev, and S. A. McCarroll. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161 (5):1202–1214, May 2015. doi: 10.1016/j.cell.2015.05.002. URL https://doi.org/10.1016/j.cell.2015.05.002.

G. L. Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, J. Fan, L. E. Borm, Z. Liu, D. van Bruggen, J. Guo, X. He, R. Barker, E. Sundström, G. Castelo-Branco, P. Cramer, I. Adameyko, S. Linnarsson, and P. V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, Aug. 2018. doi: 10.1038/s41586-018-0414-6. URL https://doi.org/10.1038/s41586-018-0414-6.

T. Masuda, R. Sankowski, O. Staszewski, C. Böttcher, L. Amann, Sagar, C. Scheiwe, S. Nessler, P. Kunz, G. van Loo, V. A. Coenen, P. C. Reinacher, A. Michel, U. Sure, R. Gold, D. Grün, J. Priller, C. Stadelmann, and M. Prinz. Spatial and temporal heterogeneity of mouse and human microglia at single-cell resolution. *Nature*, 566(7744):388–392, Feb. 2019. doi: 10.1038/s41586-019-0924-x. URL https://doi.org/10.1038/s41586-019-0924-x.

E. Mathieu, C. Le Lan, C. J. Maddison, R. Tomioka, and Y. Whye Teh. Hierarchical Representations with Poincar\'e Variational Auto-Encoders. *arXiv*, art. arXiv:1901.06033, Jan 2019.

H. Matsumoto, H. Kiryu, C. Furusawa, M. S. H. Ko, S. B. H. Ko, N. Gouda, T. Hayashi, and I. Nikaido. SCODE: an efficient regulatory network inference algorithm from single-cell RNA-seq during differentiation. *Bioinformatics*, 33(15):2314–2321, Apr. 2017. doi: 10.1093/bioinformatics/btx194. URL https://doi.org/10.1093/bioinformatics/btx194.

A. McDavid, G. Finak, and R. Gottardo. The contribution of cell cycle to heterogeneity in single-cell RNA-seq data. *Nature Biotechnology*, 34(6):591–593, June 2016. doi: 10.1038/nbt.3498. URL https://doi.org/10.1038/nbt.3498.

W. A. McGee, H. Pimentel, L. Pachter, and J. Y. Wu. Compositional data analysis is necessary for simulating and analyzing RNA-seq data. *bioRxiv*, Mar. 2019. doi: 10.1101/564955. URL https://doi.org/10.1101/564955.

W. McKinney. Data structures for statistical computing in python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

A. McLysaght and D. Guerzoni. New genes from non-coding sequence: the role of de novo protein-coding genes in eukaryotic evolutionary innovation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1678): 20140332, Aug. 2015. doi: 10.1098/rstb.2014.0332. URL https://doi.org/10.1098/rstb.2014.0332.

A. McLysaght and L. D. Hurst. Open questions in the study of de novo genes: what, how and why. *Nature Reviews Genetics*, 17(9):567–578, July 2016. doi: 10.1038/nrg.2016.78. URL https://doi.org/10.1038/nrg.2016.78.

E. Mereu, A. Lafzi, C. Moutinho, C. Ziegenhain, D. J. MacCarthy, A. Alvarez, E. Batlle, Sagar, D. Grün, J. K. Lau, S. C. Boutet, C. Sanada, A. Ooi, R. C. Jones, K. Kaihara, C. Brampton, Y. Talaga, Y. Sasagawa, K. Tanaka, T. Hayashi,

I. Nikaido, C. Fischer, S. Sauer, T. Trefzer, C. Conrad, X. Adiconis, L. T. Nguyen, A. Regev, J. Z. Levin, S. Parekh, A. Janjic, L. E. Wange, J. W. Bagnoli, W. Enard, M. Gut, R. Sandberg, I. Gut, O. Stegle, and H. Heyn. Benchmarking single-cell RNA sequencing protocols for cell atlas projects. *bioRxiv*, May 2019. doi: 10.1101/630087. URL `https://doi.org/10.1101/630087`.

M. Mikulski and J. Duda. Toroidal AutoEncoder. *arXiv*, art. arXiv:1903.12286, Mar 2019.

E. P. Mimitou, A. Cheng, A. Montalbano, S. Hao, M. Stoeckius, M. Legut, T. Roush, A. Herrera, E. Papalexi, Z. Ouyang, R. Satija, N. E. Sanjana, S. B. Koralov, and P. Smibert. Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nature Methods*, 16(5):409–412, Apr. 2019. doi: 10.1038/s41592-019-0392-0. URL `https://doi.org/10.1038/s41592-019-0392-0`.

J. Misra. Interactive exploration of microarray gene expression patterns in a reduced dimensional space. *Genome Research*, 12(7):1112–1120, July 2002. doi: 10.1101/gr.225302. URL `https://doi.org/10.1101/gr.225302`.

S. Mohammadi, J. Davila-Velderrain, M. Kellis, and A. Grama. DECODE-ing sparsity patterns in single-cell RNA-seq. *bioRxiv*, Jan. 2018. doi: 10.1101/241646. URL `https://doi.org/10.1101/241646`.

V. Moignard and B. Göttgens. Dissecting stem cell differentiation using single cell expression profiling. *Current Opinion in Cell Biology*, 43:78–86, Dec. 2016. doi:

10.1016/j.ceb.2016.08.005. URL https://doi.org/10.1016/j.ceb.2016.08.005.

V. Moignard, S. Woodhouse, L. Haghverdi, A. J. Lilly, Y. Tanaka, A. C. Wilkinson, F. Buettner, I. C. Macaulay, W. Jawaid, E. Diamanti, S.-I. Nishikawa, N. Piterman, V. Kouskoff, F. J. Theis, J. Fisher, and B. Göttgens. Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature Biotechnology*, 33(3):269–276, Feb. 2015. doi: 10.1038/nbt.3154. URL https://doi.org/10.1038/nbt.3154.

G. Morel, L. Sterck, D. Swennen, M. Marcet-Houben, D. Onesime, A. Levasseur, N. Jacques, S. Mallet, A. Couloux, K. Labadie, J. Amselem, J.-M. Beckerich, B. Henrissat, Y. V. de Peer, P. Wincker, J.-L. Souciet, T. Gabaldón, C. R. Tinsley, and S. Casaregola. Differential gene retention as an evolutionary mechanism to generate biodiversity and adaptation in yeasts. *Scientific Reports*, 5(1), June 2015. doi: 10.1038/srep11571. URL https://doi.org/10.1038/srep11571.

N. Moris, C. Pina, and A. M. Arias. Transition states and cell fate decisions in epigenetic landscapes. *Nature Reviews Genetics*, 17(11):693–703, Sept. 2016. doi: 10.1038/nrg.2016.98. URL https://doi.org/10.1038/nrg.2016.98.

A. Muscoloni and C. V. Cannistraci. Angular separability of data clusters or network communities in geometrical space and its relevance to hyperbolic embedding. *arXiv*, abs/1907.00025, 2019. URL http://arxiv.org/abs/1907.00025.

R. Neme and D. Tautz. Fast turnover of genome transcription across evolutionary time exposes entire non-coding DNA to de novo gene emergence. *eLife*, 5:09977, Feb. 2016. doi: 10.7554/elife.09977. URL https://doi.org/10.7554/elife.09977.

R. Neme, C. Amador, B. Yildirim, E. McConnell, and D. Tautz. Random sequences are an abundant source of bioactive RNAs or peptides. *Nature Ecology & Evolution*, 1(6):0127, Apr. 2017. doi: 10.1038/s41559-017-0127. URL https://doi.org/10.1038/s41559-017-0127.

E. C. Neto. Detecting learning vs memorization in deep neural networks using shared structure validation sets. *arXiv*, abs/1802.07714, 2018. URL http://arxiv.org/abs/1802.07714.

J. R. S. Newman, S. Ghaemmaghami, J. Ihmels, D. K. Breslow, M. Noble, J. L. DeRisi, and J. S. Weissman. Single-cell proteomic analysis of s. cerevisiae reveals the architecture of biological noise. *Nature*, 441(7095):840–846, May 2006. doi: 10.1038/nature04785. URL https://doi.org/10.1038/nature04785.

S. Nowotschin, M. Setty, Y.-Y. Kuo, V. Liu, V. Garg, R. Sharma, C. S. Simon, N. Saiz, R. Gardner, S. C. Boutet, D. M. Church, P. A. Hoodless, A.-K. Hadjantonakis, and D. Pe'er. The emergent landscape of the mouse gut endoderm at single-cell resolution. *Nature*, 569(7756):361–367, Apr. 2019. doi: 10.1038/s41586-019-1127-1. URL https://doi.org/10.1038/s41586-019-1127-1.

D. J. Obbard, J. Maclennan, K.-W. Kim, A. Rambaut, P. M. O'Grady, and F. M. Jiggins. Estimating divergence dates and substitution rates in the drosophila phylogeny. *Molecular Biology and Evolution*, 29(11):3459–3473, Aug. 2012. doi: 10.1093/molbev/mss150. URL https://doi.org/10.1093/molbev/mss150.

S. Ohno. Ancient linkage groups and frozen accidents. *Nature*, 244(5414):259–262, Aug. 1973. doi: 10.1038/244259a0. URL https://doi.org/10.1038/244259a0.

N. A. O'Leary, M. W. Wright, J. R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, A. Astashyn, A. Badretdin, Y. Bao, O. Blinkova, V. Brover, V. Chetvernin, J. Choi, E. Cox, O. Ermolaeva, C. M. Farrell, T. Goldfarb, T. Gupta, D. Haft, E. Hatcher, W. Hlavina, V. S. Joardar, V. K. Kodali, W. Li, D. Maglott, P. Masterson, K. M. McGarvey, M. R. Murphy, K. O'Neill, S. Pujar, S. H. Rangwala, D. Rausch, L. D. Riddick, C. Schoch, A. Shkeda, S. S. Storz, H. Sun, F. Thibaud-Nissen, I. Tolstoy, R. E. Tully, A. R. Vatsan, C. Wallin, D. Webb, W. Wu, M. J. Landrum, A. Kimchi, T. Tatusova, M. DiCuccio, P. Kitts, T. D. Murphy, and K. D. Pruitt. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, Nov. 2015. doi: 10.1093/nar/gkv1189. URL https://doi.org/10.1093/nar/gkv1189.

S. C. Olhede and P. J. Wolfe. The future of statistics and data science. *Statistics &*

Probability Letters, 136:46–50, May 2018. doi: 10.1016/j.spl.2018.02.042. URL https://doi.org/10.1016/j.spl.2018.02.042.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

F. Paul, Y. Arkin, A. Giladi, D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, D. Winter, D. Lara-Astiaso, M. Gury, A. Weiner, E. David, N. Cohen, F. K. B. Lauridsen, S. Haas, A. Schlitzer, A. Mildner, F. Ginhoux, S. Jung, A. Trumpp, B. T. Porse, A. Tanay, and I. Amit. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, 163(7):1663–1677, Dec. 2015. doi: 10.1016/j.cell.2015.11.013. URL https://doi.org/10.1016/j.cell.2015.11.013.

W. R. Pearson. Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.*, 132:185–219, 2000.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

T. Peng, Q. Zhu, P. Yin, and K. Tan. SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. *Genome Biology*, 20(1), May 2019.

doi: 10.1186/s13059-019-1681-8. URL https://doi.org/10.1186/s13059-019-1681-8.

E. Pierson and C. Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1), Nov. 2015. doi: 10.1186/s13059-015-0805-z. URL https://doi.org/10.1186/s13059-015-0805-z.

P. Portin and A. Wilkins. The evolving definition of the term "gene". *Genetics*, 205 (4):1353–1364, Mar. 2017. doi: 10.1534/genetics.116.196956. URL https://doi.org/10.1534/genetics.116.196956.

S. Prabhakaran, E. Azizi, A. Carr, and D. Pe'er. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1070–1079. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045504.

A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. M. Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *bioRxiv*, May 2019. doi: 10.1101/642926. URL https://doi.org/10.1101/642926.

T. Qin, C. mei Fan, T. zhang Wang, L. Yang, W. liang Shen, H. Sun, J. xin Lin, M. Cucchiarini, N. D. Clement, C. E. Mason, V. Bunpetch, N. Nakamura, R. Bhonde, N. D. Clement, Z. Yin, and X. Chen. Single-cell RNA-seq reveals

novel mitochondria-related musculoskeletal cell populations during adult axolotl limb regeneration process. *bioRxiv*, July 2019. doi: 10.1101/704841. URL https://doi.org/10.1101/704841.

P. Qiu. Embracing the dropouts in single-cell RNA-seq data. *bioRxiv*, Nov. 2018. doi: 10.1101/468025. URL https://doi.org/10.1101/468025.

X. Qiu, A. Hill, J. Packer, D. Lin, Y.-A. Ma, and C. Trapnell. Single-cell mRNA quantification and differential analysis with census. *Nature Methods*, 14(3):309–315, Jan. 2017a. doi: 10.1038/nmeth.4150. URL https://doi.org/10.1038/nmeth.4150.

X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods*, 14(10):979–982, Aug. 2017b. doi: 10.1038/nmeth.4402. URL https://doi.org/10.1038/nmeth.4402.

T. P. Quinn, M. F. Richardson, D. Lovell, and T. M. Crowley. propr: An r-package for identifying proportionally abundant features using compositional data analysis. *Scientific Reports*, 7(1), Nov. 2017. doi: 10.1038/s41598-017-16520-0. URL https://doi.org/10.1038/s41598-017-16520-0.

A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv*, art. arXiv:1511.06434, Nov 2015.

A. Raj and A. van Oudenaarden. Nature, nurture, or chance: Stochastic gene

expression and its consequences. *Cell*, 135(2):216–226, Oct. 2008. doi: 10.1016/j.cell.2008.09.050. URL https://doi.org/10.1016/j.cell.2008.09.050.

V. Ramani, X. Deng, R. Qiu, C. Lee, C. M. Disteche, W. S. Noble, J. Shendure, and Z. Duan. Sci-hi-c: A single-cell hi-c method for mapping 3d genome organization in large number of single cells. *Methods*, 170:61–68, Jan. 2020. doi: 10.1016/j.ymeth.2019.09.012. URL https://doi.org/10.1016/j.ymeth.2019.09.012.

L. Rampášek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg. Dr.VAE: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, Mar. 2019. doi: 10.1093/bioinformatics/btz158. URL https://doi.org/10.1093/bioinformatics/btz158.

J. M. Raser. Noise in gene expression: Origins, consequences, and control. *Science*, 309(5743):2010–2013, Sept. 2005. doi: 10.1126/science.1105891. URL https://doi.org/10.1126/science.1105891.

S. Rashid, S. Shah, Z. Bar-Joseph, and R. Pandya. Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. *Bioinformatics*, Feb. 2019. doi: 10.1093/bioinformatics/btz095. URL https://doi.org/10.1093/bioinformatics/btz095.

A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, H. Clevers, B. Deplancke,

I. Dunham, J. Eberwine, R. Eils, W. Enard, A. Farmer, L. Fugger, B. Göttgens, N. Hacohen, M. Haniffa, M. Hemberg, S. Kim, P. Klenerman, A. Kriegstein, E. Lein, S. Linnarsson, E. Lundberg, J. Lundeberg, P. Majumder, J. C. Marioni, M. Merad, M. Mhlanga, M. Nawijn, M. Netea, G. Nolan, D. Pe'er, A. Phillipakis, C. P. Ponting, S. Quake, W. Reik, O. Rozenblatt-Rosen, J. Sanes, R. Satija, T. N. Schumacher, A. Shalek, E. Shapiro, P. Sharma, J. W. Shin, O. Stegle, M. Stratton, M. J. T. Stubbington, F. J. Theis, M. Uhlen, A. van Oudenaarden, A. Wagner, F. Watt, J. Weissman, B. Wold, R. Xavier, and N. Y. and. The human cell atlas, Dec. 2017. URL `https://doi.org/10.7554/elife.27041`.

J. A. Reinhardt, B. M. Wanjiru, A. T. Brant, P. Saelao, D. J. Begun, and C. D. Jones. De novo ORFs in drosophila are important to organismal fitness and evolved rapidly from previously non-coding sequences. *PLoS Genetics*, 9(10):e1003860, Oct. 2013. doi: 10.1371/journal.pgen.1003860. URL `https://doi.org/10.1371/journal.pgen.1003860`.

M. Remmert, A. Biegert, A. Hauser, and J. Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2): 173–175, Dec. 2011. doi: 10.1038/nmeth.1818. URL `https://doi.org/10.1038/nmeth.1818`.

A. Richard, L. Boullu, U. Herbach, A. Bonnafoux, V. Morin, E. Vallin, A. Guillemin, N. P. Gao, R. Gunawan, J. Cosette, O. Arnaud, J.-J. Kupiec, T. Espinasse, S. Gonin-Giraud, and O. Gandrillon. Single-cell-based analysis highlights a surge in cell-to-cell molecular variability preceding irreversible commit-

ment in a differentiation process. *PLOS Biology*, 14(12):e1002585, Dec. 2016. doi: 10.1371/journal.pbio.1002585. URL https://doi.org/10.1371/journal.pbio.1002585.

G. M. Richardson, J. Lannigan, and I. G. Macara. Does FACS perturb gene expression? *Cytometry Part A*, 87(2):166–175, Jan. 2015. doi: 10.1002/cyto.a.22608. URL https://doi.org/10.1002/cyto.a.22608.

D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9(1), Jan. 2018. doi: 10.1038/s41467-017-02554-5. URL https://doi.org/10.1038/s41467-017-02554-5.

M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25, 2010. doi: 10.1186/gb-2010-11-3-r25. URL https://doi.org/10.1186/gb-2010-11-3-r25.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, Nov. 2009. doi: 10.1093/bioinformatics/btp616. URL https://doi.org/10.1093/bioinformatics/btp616.

S. G. Rodriques, R. R. Stickels, A. Goeva, C. A. Martin, E. Murray, C. R. Vanderburg, J. Welch, L. M. Chen, F. Chen, and E. Z. Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Sci-*

ence, 363(6434):1463–1467, Mar. 2019. doi: 10.1126/science.aaw1219. URL https://doi.org/10.1126/science.aaw1219.

T. Ronan, Z. Qi, and K. M. Naegle. Avoiding common pitfalls when clustering biological data. *Science Signaling*, 9(432):re6–re6, June 2016. doi: 10.1126/scisignal.aad1932. URL https://doi.org/10.1126/scisignal.aad1932.

J. Ronen and A. Akalin. netSmooth: Network-smoothing based imputation for single cell RNA-seq. *F1000Research*, 7:8, July 2018. doi: 10.12688/f1000research.13511.3. URL https://doi.org/10.12688/f1000research.13511.3.

K. Rooijers, C. M. Markodimitraki, F. J. Rang, S. S. de Vries, A. Chialastri, K. L. de Luca, D. Mooijman, S. S. Dey, and J. Kind. Simultaneous quantification of protein–DNA contacts and transcriptomes in single cells. *Nature Biotechnology*, 37(7):766–772, June 2019. doi: 10.1038/s41587-019-0150-y. URL https://doi.org/10.1038/s41587-019-0150-y.

J. Ruiz-Orera, J. Hernandez-Rodriguez, C. Chiva, E. Sabidó, I. Kondova, R. Bontrop, T. Marqués-Bonet, and M. Albà. Origins of de novo genes in human and chimpanzee. *PLOS Genetics*, 11(12):e1005721, Dec. 2015. doi: 10.1371/journal.pgen.1005721. URL https://doi.org/10.1371/journal.pgen.1005721.

J. Ruiz-Orera, P. Verdaguer-Grau, J. L. Villanueva-Cañas, X. Messeguer, and

M. M. Albà. Translation of neutrally evolving peptides provides a basis for de novo gene evolution. *Nature Ecology & Evolution*, 2(5):890–896, Mar. 2018. doi: 10.1038/s41559-018-0506-6. URL https://doi.org/10.1038/s41559-018-0506-6.

W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, Apr. 2019. doi: 10.1038/s41587-019-0071-9. URL https://doi.org/10.1038/s41587-019-0071-9.

M. Saint, F. Bertaux, W. Tang, X.-M. Sun, L. Game, A. Köferle, J. Bähler, V. Shahrezaei, and S. Marguerat. Single-cell imaging and RNA sequencing reveal patterns of gene expression heterogeneity during fission yeast growth and adaptation. *Nature Microbiology*, 4(3):480–491, Feb. 2019. doi: 10.1038/s41564-018-0330-4. URL https://doi.org/10.1038/s41564-018-0330-4.

C. Savojardo, P. L. Martelli, P. Fariselli, G. Profiti, and R. Casadio. BUSCA: an integrative web server to predict subcellular localization of proteins. *Nucleic Acids Research*, 46(W1):W459–W466, Apr. 2018. doi: 10.1093/nar/gky320. URL https://doi.org/10.1093/nar/gky320.

A. A. Schaffer. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Research*, 29(14):2994–3005, July 2001. doi: 10.1093/nar/29.14.2994. URL https://doi.org/10.1093/nar/29.14.2994.

C. Schlötterer. Genes from scratch – the evolutionary fate of de novo genes. *Trends in Genetics*, 31(4):215–219, Apr. 2015. doi: 10.1016/j.tig.2015.02.007. URL https://doi.org/10.1016/j.tig.2015.02.007.

A. Scialdone, K. N. Natarajan, L. R. Saraiva, V. Proserpio, S. A. Teichmann, O. Stegle, J. C. Marioni, and F. Buettner. Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods*, 85:54–61, Sept. 2015. doi: 10.1016/j.ymeth.2015.06.021. URL https://doi.org/10.1016/j.ymeth.2015.06.021.

S. Siebert, J. A. Farrell, J. F. Cazet, Y. Abeykoon, A. S. Primack, C. E. Schnitzler, and C. E. Juliano. Stem cell differentiation trajectories in hydra resolved at single-cell resolution. *Science*, 365(6451):eaav9314, July 2019. doi: 10.1126/science.aav9314. URL https://doi.org/10.1126/science.aav9314.

N. Siew and D. Fischer. Analysis of singleton ORFans in fully sequenced microbial genomes. *Proteins: Structure, Function, and Genetics*, 53(2):241–251, Sept. 2003. doi: 10.1002/prot.10423. URL https://doi.org/10.1002/prot.10423.

M. A. Skinnider, J. W. Squair, and L. J. Foster. Evaluating measures of association for single-cell transcriptomics. *Nature Methods*, 16(5):381–386, Apr. 2019. doi: 10.1038/s41592-019-0372-4. URL https://doi.org/10.1038/s41592-019-0372-4.

T. Smith, A. Heger, and I. Sudbery. UMI-tools: modeling sequencing errors in

unique molecular identifiers to improve quantification accuracy. *Genome Research*, 27(3):491–499, Jan. 2017. doi: 10.1101/gr.209601.116. URL https://doi.org/10.1101/gr.209601.116.

C. Soneson and M. D. Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods*, 15(4):255–261, Feb. 2018. doi: 10.1038/nmeth.4612. URL https://doi.org/10.1038/nmeth.4612.

S. W. Soukup. Evolution by gene duplication. s. ohno. springer-verlag, new york. 1970. 160 pp. *Teratology*, 9(2):250–251, Apr. 1974. doi: 10.1002/tera.1420090224. URL https://doi.org/10.1002/tera.1420090224.

A. Srivastava, L. Malik, T. Smith, I. Sudbery, and R. Patro. Alevin efficiently estimates accurate gene abundances from dscRNA-seq data. *Genome Biology*, 20 (1), Mar. 2019. doi: 10.1186/s13059-019-1670-y. URL https://doi.org/10.1186/s13059-019-1670-y.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

R. Stadhouders, G. J. Filion, and T. Graf. Transcription factors and 3d genome conformation in cell-fate decisions. *Nature*, 569(7756):345–354, May 2019. doi: 10.1038/s41586-019-1182-7. URL https://doi.org/10.1038/s41586-019-1182-7.

J. S. Stanley, G. Wolf, and S. Krishnaswamy. Manifold alignment with feature correspondence. *arXiv*, abs/1810.00386, 2018. URL http://arxiv.org/abs/1810.00386.

R. Stark, M. Grzelak, and J. Hadfield. RNA sequencing: the teenage years. *Nature Reviews Genetics*, July 2019. doi: 10.1038/s41576-019-0150-2. URL https://doi.org/10.1038/s41576-019-0150-2.

Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson. Big data: Astronomical or genomical? *PLOS Biology*, 13(7):e1002195, July 2015. doi: 10.1371/journal.pbio.1002195. URL https://doi.org/10.1371/journal.pbio.1002195.

N. B. Stewart and R. L. Rogers. Chromosomal rearrangements as a source of new gene formation in drosophila yakuba. *PLOS Genetics*, 15(9):e1008314, Sept. 2019. doi: 10.1371/journal.pgen.1008314. URL https://doi.org/10.1371/journal.pgen.1008314.

M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868, July 2017. doi: 10.1038/nmeth.4380. URL https://doi.org/10.1038/nmeth.4380.

K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Du-

doit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(1), June 2018. doi: 10.1186/s12864-018-4772-0. URL https://doi.org/10.1186/s12864-018-4772-0.

L. Struski, J. Tabor, I. T. Podolak, and A. Nowak. Interpolation in generative models. *arXiv*, abs/1904.03445, 2019. URL http://arxiv.org/abs/1904.03445.

T. Stuart and R. Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, Jan. 2019. doi: 10.1038/s41576-019-0093-7. URL https://doi.org/10.1038/s41576-019-0093-7.

T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, June 2019. doi: 10.1016/j.cell.2019.05.031. URL https://doi.org/10.1016/j.cell.2019.05.031.

W. Sun, X.-W. Zhao, and Z. Zhang. Identification and evolution of the orphan genes in the domestic silkworm, bombyx mori. *FEBS Letters*, 589(19PartB): 2731–2738, Aug. 2015. doi: 10.1016/j.febslet.2015.08.008. URL https://doi.org/10.1016/j.febslet.2015.08.008.

V. Svensson and E. da Veiga Beltrame. A curated database reveals trends in single cell transcriptomics. *bioRxiv*, Aug. 2019. doi: 10.1101/742304. URL https://doi.org/10.1101/742304.

V. Svensson, K. N. Natarajan, L.-H. Ly, R. J. Miragaia, C. Labalette, I. C. Macaulay,

A. Cvejic, and S. A. Teichmann. Power analysis of single-cell RNA-sequencing experiments. *Nature Methods*, 14(4):381–387, Mar. 2017. doi: 10.1038/nmeth.4220. URL https://doi.org/10.1038/nmeth.4220.

C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv*, abs/1602.07261, 2016. URL http://arxiv.org/abs/1602.07261.

T. Tabib, C. Morse, T. Wang, W. Chen, and R. Lafyatis. SFRP2/DPP4 and FMO1/LSP1 define major fibroblast populations in human skin. *Journal of Investigative Dermatology*, 138(4):802–810, Apr. 2018. doi: 10.1016/j.jid.2017.09.045. URL https://doi.org/10.1016/j.jid.2017.09.045.

J. TAN, M. UNG, C. CHENG, and C. S. GREENE. UNSUPERVISED FEATURE CONSTRUCTION AND KNOWLEDGE EXTRACTION FROM GENOME-WIDE ASSAYS OF BREAST CANCER WITH DENOISING AUTOENCODERS. In *Biocomputing 2015*. WORLD SCIENTIFIC, Nov. 2014. doi: 10.1142/9789814644730_0014. URL https://doi.org/10.1142/9789814644730_0014.

J. Tan, J. H. Hammond, D. A. Hogan, and C. S. Greene. ADAGE-based integration of publicly available pseudomonas aeruginosa gene expression data with denoising autoencoders illuminates microbe-host interactions. *mSystems*, 1(1), Jan. 2016. doi: 10.1128/msystems.00025-15. URL https://doi.org/10.1128/msystems.00025-15.

J. Tan, G. Doing, K. A. Lewis, C. E. Price, K. M. Chen, K. C. Cady, B. Perchuk, M. T. Laub, D. A. Hogan, and C. S. Greene. Unsupervised extraction of stable expression signatures from public compendia with an ensemble of neural networks. *Cell Systems*, 5(1):63–71.e6, July 2017. doi: 10.1016/j.cels.2017.06.003. URL https://doi.org/10.1016/j.cels.2017.06.003.

A. Tanay and A. Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, Jan. 2017. doi: 10.1038/nature21350. URL https://doi.org/10.1038/nature21350.

F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, K. Lao, and M. A. Surani. mRNA-seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, Apr. 2009. doi: 10.1038/nmeth.1315. URL https://doi.org/10.1038/nmeth.1315.

C. Targonski, B. T. Shealy, M. C. Smith, and F. A. Feltus. Cellular state transformations using generative adversarial networks. *arXiv*, abs/1907.00118, 2019. URL http://arxiv.org/abs/1907.00118.

M. Toll-Riera, N. Bosch, N. Bellora, R. Castelo, L. Armengol, X. Estivill, and M. M. Alba. Origin of primate orphan genes: A comparative genomics approach. *Molecular Biology and Evolution*, 26(3):603–612, Dec. 2008. doi: 10.1093/molbev/msn281. URL https://doi.org/10.1093/molbev/msn281.

S. Tracy, G.-C. Yuan, and R. Dries. RESCUE: imputing dropout events in

single-cell RNA-sequencing data. *BMC Bioinformatics*, 20(1), July 2019. doi: 10.1186/s12859-019-2977-0. URL https://doi.org/10.1186/s12859-019-2977-0.

C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, Mar. 2014. doi: 10.1038/nbt.2859. URL https://doi.org/10.1038/nbt.2859.

V. Tretyachenko, J. Vymětal, L. Bednárová, V. Kopecký, K. Hofbauerová, H. Jindrová, M. Hubálek, R. Souček, J. Konvalinka, J. Vondrášek, and K. Hlouchová. Random protein sequences can form defined secondary structures and are well-tolerated in vivo. *Scientific Reports*, 7(1):15449, Nov. 2017. doi: 10.1038/s41598-017-15635-8. URL https://doi.org/10.1038/s41598-017-15635-8.

T. N. Trong, R. Kramer, J. Mehtonen, G. González, V. Hautamäki, and M. Heinäniemi. SISUA: Semi-supervised generative autoencoder for single cell data. *bioRxiv*, May 2019. doi: 10.1101/631382. URL https://doi.org/10.1101/631382.

N. Vakirlis and A. McLysaght. Computational prediction of de novo emerged protein-coding genes. In *Methods in Molecular Biology*, pages 63–81. Springer New York, Sept. 2018. doi: 10.1007/978-1-4939-8736-8_4. URL https://doi.org/10.1007/978-1-4939-8736-8_4.

N. Vakirlis, A. S. Hebert, D. A. Opulente, G. Achaz, C. T. Hittinger, G. Fischer, J. J. Coon, and I. Lafontaine. A molecular portrait of de novo genes in yeasts. *Molecular Biology and Evolution*, 35(3):631–645, Dec. 2017. doi: 10.1093/molbev/msx315. URL https://doi.org/10.1093/molbev/msx315.

N. Vakirlis, A.-R. Carvunis, and A. McLysaght. Synteny-based analyses indicate that sequence divergence is not the main source of orphan genes. *bioRxiv*, page 735175, Aug. 2019. doi: 10.1101/735175. URL https://doi.org/10.1101/735175.

C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14(6):565–571, May 2017. doi: 10.1038/nmeth.4292. URL https://doi.org/10.1038/nmeth.4292.

K. G. van den Boogaart and R. Tolosana-Delgado. "compositions": A unified r package to analyze compositional data. *Computers & Geosciences*, 34(4):320–338, Apr. 2008. doi: 10.1016/j.cageo.2006.11.017. URL https://doi.org/10.1016/j.cageo.2006.11.017.

L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure

for efficient numerical computation. *Computing in Science Engineering*, 13(2): 22–30, March 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2011.37.

D. van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, B. Bierie, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.e27, July 2018. doi: 10.1016/j.cell.2018.05.061. URL https://doi.org/10.1016/j.cell.2018.05.061.

A. Veidenberg, A. Medlar, and A. Löytynoja. Wasabi: An integrated platform for evolutionary sequence analysis and data visualization. *Molecular Biology and Evolution*, 33(4):1126–1130, Dec. 2015. doi: 10.1093/molbev/msv333. URL https://doi.org/10.1093/molbev/msv333.

D. Velmeshev, L. Schirmer, D. Jung, M. Haeussler, Y. Perez, S. Mayer, A. Bhaduri, N. Goyal, D. H. Rowitch, and A. R. Kriegstein. Single-cell genomics identifies cell type–specific molecular changes in autism. *Science*, 364(6441):685–689, May 2019. doi: 10.1126/science.aav8130. URL https://doi.org/10.1126/science.aav8130.

K. Verboom, C. Everaert, N. Bolduc, K. J. Livak, N. Yigit, D. Rombaut, J. Anckaert, S. Lee, M. T. Venø, J. Kjems, F. Speleman, P. Mestdagh, and J. Vandesompele. SMARTer single cell total RNA sequencing. *Nucleic Acids Research*, June 2019. doi: 10.1093/nar/gkz535. URL https://doi.org/10.1093/nar/gkz535.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390294. URL http://doi.acm.org/10.1145/1390156.1390294.

T. N. Vu, Q. F. Wills, K. R. Kalari, N. Niu, L. Wang, M. Rantalainen, and Y. Pawitan. Beta-poisson model for single-cell RNA-seq data analyses. *Bioinformatics*, 32(14):2128–2135, Apr. 2016. doi: 10.1093/bioinformatics/btw202. URL https://doi.org/10.1093/bioinformatics/btw202.

F. Wagner, Y. Yan, and I. Yanai. K-nearest neighbor smoothing for high-throughput single-cell RNA-seq data. *bioRxiv*, Nov. 2017. doi: 10.1101/217737. URL https://doi.org/10.1101/217737.

D. Wang and J. Gu. VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, Proteomics & Bioinformatics*, 16(5):320–331, Oct. 2018. doi: 10.1016/j.gpb.2018.08.003. URL https://doi.org/10.1016/j.gpb.2018.08.003.

J. Wang, D. Agarwal, M. Huang, G. Hu, Z. Zhou, C. Ye, and N. R. Zhang. Data denoising with transfer learning in single-cell transcriptomics. *Nature Methods*, 16(9):875–878, Aug. 2019a. doi: 10.1038/s41592-019-0537-1. URL https://doi.org/10.1038/s41592-019-0537-1.

S. Wang, M. Karikomi, A. L. MacLean, and Q. Nie. Cell lineage and communica-

tion network inference via optimization for single-cell transcriptomics. *Nucleic Acids Research*, 47(11):e66–e66, Mar. 2019b. doi: 10.1093/nar/gkz204. URL https://doi.org/10.1093/nar/gkz204.

S. Wang, Q. Zhang, H. Hui, K. Agrawal, M. A. Y. Karris, and T. Rana. An atlas of immune cell exhaustion in HIV-infected individuals revealed by single-cell transcriptomics. *bioRxiv*, June 2019c. doi: 10.1101/678763. URL https://doi.org/10.1101/678763.

T. Wang, T. S. Johnson, W. Shao, Z. Lu, B. R. Helm, J. Zhang, and K. Huang. BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. *Genome Biology*, 20(1), Aug. 2019d. doi: 10.1186/s13059-019-1764-6. URL https://doi.org/10.1186/s13059-019-1764-6.

R. M. Waterhouse, M. Seppey, F. A. Simão, M. Manni, P. Ioannidis, G. Klioutchnikov, E. V. Kriventseva, and E. M. Zdobnov. BUSCO applications from quality assessments to gene prediction and phylogenomics. *Molecular Biology and Evolution*, 35(3):543–548, Dec. 2017. doi: 10.1093/molbev/msx319. URL https://doi.org/10.1093/molbev/msx319.

G. P. Way and C. S. Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv*, Aug. 2017. doi: 10.1101/174474. URL https://doi.org/10.1101/174474.

C. M. Weisman and S. R. Eddy. Gene evolution: Getting something from nothing.

Current Biology, 27(13):R661–R663, July 2017. doi: 10.1016/j.cub.2017.05.056. URL https://doi.org/10.1016/j.cub.2017.05.056.

J. H. Werren, S. Richards, C. A. Desjardins, O. Niehuis, J. Gadau, J. K. Colbourne, L. W. Beukeboom, C. Desplan, C. G. Elsik, C. J. P. Grimmelikhuijzen, P. Kitts, J. A. Lynch, T. Murphy, D. C. S. G. Oliveira, C. D. Smith, L. v. d. Zande, K. C. Worley, E. M. Zdobnov, M. Aerts, S. Albert, V. H. Anaya, J. M. Anzola, A. R. Barchuk, S. K. Behura, A. N. Bera, M. R. Berenbaum, R. C. Bertossa, M. M. G. Bitondi, S. R. Bordenstein, P. Bork, E. Bornberg-Bauer, M. Brunain, G. Cazzamali, L. Chaboub, J. Chacko, D. Chavez, C. P. Childers, J.-H. Choi, M. E. Clark, C. Claudianos, R. A. Clinton, A. G. Cree, A. S. Cristino, P. M. Dang, A. C. Darby, D. C. de Graaf, B. Devreese, H. H. Dinh, R. Edwards, N. Elango, E. Elhaik, O. Ermolaeva, J. D. Evans, S. Foret, G. R. Fowler, D. Gerlach, J. D. Gibson, D. G. Gilbert, D. Graur, S. Grunder, D. E. Hagen, Y. Han, F. Hauser, D. Hultmark, H. C. Hunter, G. D. D. Hurst, S. N. Jhangian, H. Jiang, R. M. Johnson, A. K. Jones, T. Junier, T. Kadowaki, A. Kamping, Y. Kapustin, B. Kechavarzi, J. Kim, J. Kim, B. Kiryutin, T. Koevoets, C. L. Kovar, E. V. Kriventseva, R. Kucharski, H. Lee, S. L. Lee, K. Lees, L. R. Lewis, D. W. Loehlin, J. M. Logsdon, J. A. Lopez, R. J. Lozado, D. Maglott, R. Maleszka, A. Mayampurath, D. J. Mazur, M. A. McClure, A. D. Moore, M. B. Morgan, J. Muller, M. C. Munoz-Torres, D. M. Muzny, L. V. Nazareth, S. Neupert, N. B. Nguyen, F. M. F. Nunes, J. G. Oakeshott, G. O. Okwuonu, B. A. Pannebakker, V. R. Pejaver, Z. Peng, S. C. Pratt, R. Predel, L.-L. Pu, H. Ranson, R. Raychoudhury, A. Rechtsteiner, J. G. Reid, M. Riddle, J. Romero-Severson, M. Rosenberg,

T. B. Sackton, D. B. Sattelle, H. Schluns, T. Schmitt, M. Schneider, A. Schuler, A. M. Schurko, D. M. Shuker, Z. L. P. Simoes, S. Sinha, Z. Smith, A. Souvorov, A. Springauf, E. Stafflinger, D. E. Stage, M. Stanke, Y. Tanaka, A. Telschow, C. Trent, S. Vattathil, L. Viljakainen, K. W. Wanner, R. M. Waterhouse, J. B. Whitfield, T. E. Wilkes, M. Williamson, J. H. Willis, F. Wolschin, S. Wyder, T. Yamada, S. V. Yi, C. N. Zecher, L. Zhang, and R. A. G. and. Functional and evolutionary insights from the genomes of three parasitoid nasonia species. *Science*, 327(5963):343–348, Jan. 2010. doi: 10.1126/science.1178028. URL `https://doi.org/10.1126/science.1178028`.

T. White. Sampling generative networks: Notes on a few effective techniques. *arXiv*, abs/1609.04468, 2016. URL `http://arxiv.org/abs/1609.04468`.

S. Willis and J. Masel. Gene birth contributes to structural disorder encoded by overlapping genes. *Genetics*, 210(1):303–313, July 2018. doi: 10.1534/genetics.118.301249. URL `https://doi.org/10.1534/genetics.118.301249`.

B. A. Wilson and J. Masel. Putatively noncoding transcripts show extensive association with ribosomes. *Genome Biology and Evolution*, 3:1245–1252, Jan. 2011. doi: 10.1093/gbe/evr099. URL `https://doi.org/10.1093/gbe/evr099`.

B. A. Wilson, S. G. Foy, R. Neme, and J. Masel. Young genes are highly disordered as predicted by the preadaptation hypothesis of de novo gene birth. *Nature Ecol-*

ogy & Evolution, 1(6):0146, Apr. 2017. doi: 10.1038/s41559-017-0146. URL https://doi.org/10.1038/s41559-017-0146.

E. M. Wissink, A. Vihervaara, N. D. Tippens, and J. T. Lis. Nascent RNA analyses: tracking transcription and its regulation. *Nature Reviews Genetics*, Aug. 2019. doi: 10.1038/s41576-019-0159-6. URL https://doi.org/10.1038/s41576-019-0159-6.

L. Wissler, J. Gadau, D. F. Simola, M. Helmkampf, and E. Bornberg-Bauer. Mechanisms and dynamics of orphan gene emergence in insect genomes. *Genome Biology and Evolution*, 5(2):439–455, Jan. 2013. doi: 10.1093/gbe/evt009. URL https://doi.org/10.1093/gbe/evt009.

F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), Feb. 2018. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

B. Wu and A. Knudson. Tracing the de novo origin of protein-coding genes in yeast. *mBio*, 9(4):0102418, July 2018. doi: 10.1128/mbio.01024-18. URL https://doi.org/10.1128/mbio.01024-18.

P. Xie, M. Gao, C. Wang, J. Zhang, P. Noel, C. Yang, D. V. Hoff, H. Han, M. Q. Zhang, and W. Lin. SuperCT: a supervised-learning framework for enhanced characterization of single-cell transcriptomic profiles. *Nucleic Acids Research*,

47(8):e48–e48, Feb. 2019. doi: 10.1093/nar/gkz116. URL https://doi.org/10.1093/nar/gkz116.

C. Xu and Z. Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980, Feb. 2015. doi: 10.1093/bioinformatics/btv088. URL https://doi.org/10.1093/bioinformatics/btv088.

Z. Yang and J. Huang. De novo origin of new genes with introns inPlasmodium vivax. *FEBS Letters*, 585(4):641–644, Jan. 2011. doi: 10.1016/j.febslet.2011.01.017. URL https://doi.org/10.1016/j.febslet.2011.01.017.

G. Yehudai and O. Shamir. On the Power and Limitations of Random Features for Understanding Neural Networks. *arXiv*, art. arXiv:1904.00687, Apr 2019.

S. H. Yip, P. Wang, J.-P. A. Kocher, P. C. Sham, and J. Wang. Linnorm: improved statistical analysis for single cell RNA-seq expression data. *Nucleic Acids Research*, 45(22):e179–e179, Sept. 2017. doi: 10.1093/nar/gkx828. URL https://doi.org/10.1093/nar/gkx828.

S. H. Yip, P. C. Sham, and J. Wang. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Briefings in Bioinformatics*, Feb. 2018. doi: 10.1093/bib/bby011. URL https://doi.org/10.1093/bib/bby011.

X.-X. Yu, W.-L. Qiu, L. Yang, Y. Zhang, M.-Y. He, L.-C. Li, and C.-R. Xu. Defining multistep cell fate decision pathways during pancreatic development at single-

cell resolution. *The EMBO Journal*, 38(8), Feb. 2019. doi: 10.15252/embj.2018100164. URL https://doi.org/10.15252/embj.2018100164.

L. Zappia, B. Phipson, and A. Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(1), Sept. 2017. doi: 10.1186/s13059-017-1305-0. URL https://doi.org/10.1186/s13059-017-1305-0.

Y. Zarai and T. Tuller. Computational analysis of the oscillatory behavior at the translation level induced by mRNA levels oscillations due to finite intracellular resources. *PLOS Computational Biology*, 14(4):e1006055, Apr. 2018. doi: 10.1371/journal.pcbi.1006055. URL https://doi.org/10.1371/journal.pcbi.1006055.

Zhang, Luo, Zhong, Choi, Ma, Wang, Mahrt, Guo, Stawiski, Modrusan, Seshagiri, Kapur, Hon, Brugarolas, and Wang. SCINA: Semi-supervised analysis of single cells in silico. *Genes*, 10(7):531, July 2019a. doi: 10.3390/genes10070531. URL https://doi.org/10.3390/genes10070531.

L. Zhang, Y. Ren, T. Yang, G. Li, J. Chen, A. R. Gschwend, Y. Yu, G. Hou, J. Zi, R. Zhou, B. Wen, J. Zhang, K. Chougule, M. Wang, D. Copetti, Z. Peng, C. Zhang, Y. Zhang, Y. Ouyang, R. A. Wing, S. Liu, and M. Long. Rapid evolution of protein diversity by de novo origination in oryza. *Nature Ecology & Evolution*, 3(4):679–690, Mar. 2019b. doi: 10.1038/s41559-019-0822-5. URL https://doi.org/10.1038/s41559-019-0822-5.

X. Zhang, C. Xu, and N. Yosef. SymSim: simulating multi-faceted variability in single cell RNA sequencing. *bioRxiv*, July 2018. doi: 10.1101/378646. URL https://doi.org/10.1101/378646.

L. Zhao, P. Saelao, C. D. Jones, and D. J. Begun. Origin and spread of de novo genes in drosophila melanogaster populations. *Science*, 343(6172):769–772, Jan. 2014. doi: 10.1126/science.1248286. URL https://doi.org/10.1126/science.1248286.

Q. Zhou, G. Zhang, Y. Zhang, S. Xu, R. Zhao, Z. Zhan, X. Li, Y. Ding, S. Yang, and W. Wang. On the origin of new genes in drosophila. *Genome Research*, 18 (9):1446–1455, July 2008. doi: 10.1101/gr.076588.108. URL https://doi.org/10.1101/gr.076588.108.

C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard. Comparative analysis of single-cell RNA sequencing methods. *Molecular Cell*, 65(4):631–643.e4, Feb. 2017. doi: 10.1016/j.molcel.2017.01.023. URL https://doi.org/10.1016/j.molcel.2017.01.023.

B. Zimmermann, N. S. M. Robert, U. Technau, and O. Simakov. Ancient animal genome architecture reflects cell type identities. *Nature Ecology & Evolution*, 3 (9):1289–1293, Aug. 2019. doi: 10.1038/s41559-019-0946-7. URL https://doi.org/10.1038/s41559-019-0946-7.