Politecnico di Torino

Master's Degree in Data Science and Engineering

Master of Science Thesis

In Silico Perturbation of Single Cells

A Spiking Protocol for Metric Evaluation

Supervisors
Prof. Francesca Buffa
Prof. Alfredo Benso
Prof. Gianfranco Politano

Candidate
Federico Borra

October 2024

Abstract

One of the main goals of computational biology is to develop realistic models of cells, such that their behaviour can be studied *in silico* (i.e. in a computer simulation) and conclusions can be drawn on the actual biological phenomena we are considering. A perturbation, in this work, is defined as a change in the external environment or in the inner mechanisms of the cell. In order to produce actionable simulations it's imperative that the response of a model to a perturbation is as close as possible to what happens in reality.

The aim of this work is to establish a metric of evaluation of different models, able to discern which among them behaves most similarly to experimental results. As we will see there is no agreed upon method in the literature, and the commonly employed strategies have some disadvantages that will be highlighted.

The contribution is twofold: firstly a *spiking protocol*, that allows us to detect the best among competing metrics. Secondly, the development of a method taking full advantage of the characteristics of single cell data, mainly the joint probability distribution of the gene expression levels, that can now be estimated and could not have been with traditional bulk transcriptomics.

With bulk transcriptomics in fact we can only determine the average expression levels of a given gene in the sample. Instead with single cell data we can appreciate the complex intertwining of the various genes' activity, since we can see for any given cell whether a certain gene tends to be co-expressed with others, and so on.

Current methods are lacking on this point since they perform evaluations by aggregating data, in what's called pseudo-bulk, i.e. averaging the expression levels for any gene in a sample sequenced with scRNA-seq. This is almost equivalent to using bulk data, therefore I argue that there's room for improvement on this front and I propose one such technique in this manuscript.

Contents

Part I

Problem Statement and Existing Work

Chapter 1

Introduction

In Part I I will explain how cells react to their environment and why gene expression is important, then I will discuss how we can investigate it with scRNA-seq, then briefly mention its limitations. Then describe in silico perturbation and some methods to perform it (Chapter 2). Lastly I will present the various existing evaluation methods in the literature (Chapter 3).

In Part II I will explain the shortcomings of the existing evaluation methods (Chapter 5), introduce my approach (Chapter 6), explaining the rationale behind it, and lastly I will describe the various mathematical and computational methods employed (Chapters 7, 8, 9).

Finally, in Part III I will present the study design I've concocted (Chapter 10), describe the datasets and computational tools used (Chapter 11), then show some results obtained (Chapter 12), finally I will present a short discussion on the work done and identify some potential future directions of inquiry (Chapter 13).

1.1 Cells Respond to their Environment

All living beings are made of cells, be they humans, plants or bacteria. As different as these organisms may be, their cells share a lot of characteristics. All cells have hereditable material in the form of DNA, some portions of DNA called genes are then read by the cell and used as blueprints to produce RNA (which sometimes is the end product). Often the RNA is then translated into proteins which are the catalysts for most processes in the cell (and sometimes have a structural function) [3].

Not all genes are active at the same time, and genes are switched on and off in response to external stimuli, through *regulatory mechanisms.* These regulatory mechanisms determine the response of the cell to the environment.

Let's see for example how the bacterium *E. coli* reacts to a change in the environment (a change caused by itself, by the way). In this example we start from a setting in which there's an abundance of both glucose and lactose, then glucose is depleted by the bacteria and only then lactose is consumed in its stead [20]. This leads us to think that glucose is *E. coli*'s preferred sugar. In case it's not available *E. coli* will digest lactose in its place. Evolutionarily it prefers it due to the higher net energy produced by breaking down glucose rather than lactose.

How can I say that *E. coli* "prefers" glucose to lactose? Does a bacterium have agency? Most likely not, a bacterium is a complex molecular machine but it cannot think for itself. It's the bacterium's *regulatory mechanisms* that determine its behavior: making it digest glucose if it's present and ignore lactose, but in case glucose runs out and lactose is present then it starts digesting lactose instead.

1.1.1 *Lac* Operon

The reason behind the phenomenon described above is the regulation of the *lac* operon. An operon is a set of genes that are transcribed in the same RNA molecule, which then gives rise to the different products coded by the genes in the operon. An organism can therefore "decide" whether to transcribe the whole operon (all the genes that comprise it) or not.

The *lac* operon contains proteins needed to digest lactose. Its transcription is regulated by two regulators: a CAP activator (that enables the transcription of the operon) and a Lac repressor (that impedes transcription).

The CAP activator responds to the presence of cyclic AMP, which is a small molecule whose concentration rises inside the bacterium when glucose is absent. The *Lac* repressor instead by default inhibits the transcription of the *Lac* operon, when lactose is present however it changes shape and allows the translation to occur [33].

Note that both conditions must occur otherwise the transcription will not take place: both glucose must be absent and lactose must be present, in a logical AND.

1.2 Regulation of Gene Expression

Mechanisms like this one are present in all kinds of cells. As a matter of fact in eukaryotic cells the complexity of regulation is considerably higher than in prokaryotes as attested also by the percentage of DNA devoted to regulatory roles [7].

The *lac* operon is a clear example of the importance of regulation and its effects, and illustrative of the interplay between genes and environment. It's not enough to study only the DNA, but we need to understand how the genome and the environment interact, through the various regulatory mechanisms.

Another thing is worth mentioning: we need not concern ourselves with the complex allosteric conformational change undergone by the *Lac* repressor and its analog in the CAP activator, if the end result is a logical AND on conditions concerning the presence or absence of some substances in the environment.

1.2.1 The Need for Computational Methods

This is good news for us as we can attempt to understand the regulatory landscape without a full knowledge of the molecular machinery behind it. Another obstacle, though, is presented by the sheer volume of possible combinations of interactions. There are more than 20,000 genes in humans [40], for instance, and every gene could potentially interact with any other (through its products), under certain conditions. The interaction can also be many-to-one as we've seen before, further complicating the problem.

This is where the need for advanced computational techniques comes in. The complexity of the problem would be insurmountable without them, but there might be hope by leveraging them.

1.3 Sequencing Revolution

In 2001 the Human Genome Project concluded a 10 year endeavor, with a cost of 3 billion dollars to obtain the sequence of a single human genome [30]. Less than 25 years later the cost reduced 3 million fold (to about 1,000 euros) and you can sequence a genome in a few days [34].

But as I explained earlier the genome by itself is not enough to understand the behavior of the cell. We need to see and model how the various genes and their product interact among themselves and with the environment.

1.3.1 RNA-seq

Here comes to the rescue the RNA sequencing technique, also called RNA-seq [50]. It consists of using a *reverse transcriptase* enzyme to convert an RNA molecule to its complementary DNA (or cDNA). The DNA can then be amplified with DNA polymerase and sequenced using the regular DNA sequencing techniques.

The advantage of this method is that it allows us to see which genes are being actively transcribed and which aren't. It also enables us to see the level of transcription of the various genes, further enhancing our ability to probe the mechanisms of the cell.

There are complications though: it's not immediate to infer the levels of a protein coded by a gene only from its RNA levels, since there could be posttranscriptional regulation. Also the proteins can be modified by the cell's internal environment, changing radically their function, as in the example of the *Lac* operon in Subsection 1.1.1.

Moreover, traditional RNA-seq, also called *bulk* RNA-seq is done on a sample of cells and basically tells us the "average" RNA levels for that sample, across the different cells. However cells are quite varied: for example a neuron and a immune cell are clearly different beasts, and averaging their gene expression levels is suboptimal if we want to understand either one or the other. The difference in cell type need not be so stark, and the same cell at different stages of its life cycle presents some variation in the genes it expresses. To address these problems nowadays its becoming increasingly more common to perform single cell RNA sequencing, or scRNA-seq.

1.3.2 scRNA-seq

Single cell RNA sequencing, or scRNA-seq is RNA-sequencing performed on a single cell at a time, enabling us to see what genes (and at which level) are being expressed by that cell [49]. This makes us able to better appreciate individual differences, and characterize subtypes (or clusters) of cells.

Since the starting genetic material from any given cell is very little, a lot of amplification is required. It is not possible to guarantee an even distribution of *reverse transcriptase* or *DNA polymerase* on a sample so small. This causes a lot of noise in the data.

Another big problem is that differences in how a batch of cells is treated, sequencing technique employed and other factors make data collected by different research groups, or even by the same group at different times, not commensurable with other data pertaining to the same phenomenon. This problem is known as *batch effect*, and it was already a problem with the earlier microarray techniques, and even more so with single cell data [42].

1.4 Perturbation Studies and CRISPR

In order to understand how cells work we need to see how they respond to perturbations, defined as changes in the external environment, or the cell's internal mechanisms.

For what concerns changes in the external environment, these are relatively easier to achieve. We can change the amount, or type, of nutrient a cell culture is in with ease. Hypoxic chambers are present in most big cancer laboratories to study how cells react to the absence of oxygen, and most importantly how they behave once re-oxygenated.

A more tricky predicament is the one in which we want to change the internal mechanisms of the cell. If we want to understand whether a gene regulates another it would be nice to be able to turn off the first and see if the second is still being transcribed or not.

Luckily the novel gene editing technique **CRISPR** [39] allows us to do exactly that, in a more streamlined and cheap way than ever before.

1.4.1 CRISPR-Cas9

CRISPR is a revolutionary gene editing mechanism based on the CRISPR-Cas9 bacterial immune system component [24].

CRISPR, or clustered regularly interspaced short palindromic repeats, are DNA sequences that are used by bacteria as a kind of antibody for bacteriophages (a type of virus that, as the name suggests, “eats” bacteria).

When a bacterium is infected a sequence from a bacteriophage DNA is captured and conserved in the bacterial DNA. When a similar phage infects the bacterium the Cas9 protein is loaded with the “antibody” sequence for that phage so it binds and cuts its DNA, impeding its functioning [54].

The main advantage of this molecular machinery is that is kind of “programmable”. Meaning that by changing the so called guide RNA the Cas9 enzyme cuts (therefore damages) different DNA sequences. This allows us to change the target genes by simply changing the guide RNA. Whereas before a whole new protein would have to be designed in a costly and lengthy process in order to edit a target gene, for example with ZFNs or TALENs [22].

Having a cheap and fast way to edit the genome of living cells allowed us to have access to a deluge of incredibly invaluable data that could enable us to understand how the various genes and their products interact with each other, and what happens when a gene is turned off.

Chapter 2

In Silico Perturbation

In silico perturbation refers to any computational method used to simulate the response of a cell to a perturbation. By simulating single cells individually, then analyzing the results as a whole we can gain insights on the whole landscape of cell types and subtypes, and better characterize their behavior.

Here I present 3 different methodologies employed to perform in silico perturbation. For each of them I will pick a representative whose metric I will evaluate in Part III.

2.1 Gene Regulatory Networks

Gene regulatory network (GRN) inference methods are, among the ones presented, the technique that most closely tries to adhere to the underlying biological reality. A review of the most up-to-date methods can be found at [5].

These kind of methods attempt to recover the regulatory relationships between genes, starting from all the possible gene-gene interactions and consecutively narrowing it down through various kind of data. The rationale behind this strategy that by using any single data type on its own we would incur in a lot of false positives but not too much false negatives, so these data sources lend themselves particularly well to this treatment. The vast majority of these methods makes use of three different data sources: transcription factor (TF) binding motifs databases, chromatin accessibility data (ATAC-seq), and lastly transcriptomic data, typically scRNA-seq, so that different GRNs can be inferred depending on the cell type (or rather cluster, usually).

2.1.1 Dictys

Dictys [48] is a GRN inference method that makes ample use of probabilistic programming, through the Pyro framework [10]. It follows the strategy of successive filtering of possible TF-gene regulation as I've described above. It recovers a dynamic gene regulatory network and models the transcriptional activity through the use of stochastic differential equations.

2.2 VAE-based

Methods based on variational autoencoders (VAEs, introduced in [25]). The way they work is to encode every cell in a lower dimensional space, then to apply latent space vector arithmetic to perturb them, and then decode them to get back to the original data, usually transcriptomic in nature. Some notable examples are scGen [32] and its evolution, CPA [31].

2.2.1 CPA

In order to learn how to predict perturbations the authors follow a procedure reminiscent of generative adversarial network (GAN) training [19]. They learn an embedding of the perturbed cells (through the encoder network) such that a discriminator network is not able to tell which perturbation was it from. Then they add the perturbation vector in the latent space, and lastly, with a decoder network, they recover the original cells' gene expression levels. In order to obtain a double or triple perturbation the respective vectors are summed in the latent space.

2.3 Transformer-based

Driven by the astounding transfer learning [38] and contextual understanding [46] [6] capabilities of transformer based models, the computational biology community is trying to replicate the revolution undergone by the natural language processing field. A review on the progress in this respect can be found at [45].

2.3.1 scGPT

scGPT [15] is a transformer-based multi-purpose single cell transcriptomics model. Despite what the name seems to suggest it is an encoder-only model (more akin to BERT [16]) and not decoder-only like GPT [38].

Every gene is a token, in which there's and embedding of its id and its expression levels. There are also special tokens to indicate the perturbation conditions and pathways.

Some tokens are then masked and the pre-training objective is the classic masked language modelling.

Chapter 3

Existing Evaluation Methods

Let us now examine how performance evaluation has been done in the literature. First I will recall some useful concepts (namely some "correlation" measures), then I will present the various methods used in the papers I have selected as representative, or state-of-the-art.

3.1 Correlation Measures

3.1.1 R-squared

The coefficient of determination (or R^2) is not a correlation measure, but it's similar enough to one for our purposes.

Definition 1 **(R^2)** *Let the sum of squares of residuals be*

$$\mathrm{SS}_{res} \doteq \sum_i (y_i - f(x_i))^2 \tag{3.1}$$

Let the total sum of squares be

$$\mathrm{SS}_{tot} \doteq \sum_i (y_i - \bar{y})^2 \tag{3.2}$$

Then the coefficient of determination R^2 is defined as

$$R^2 \doteq 1 - \frac{\mathrm{SS}_{res}}{\mathrm{SS}_{tot}} \tag{3.3}$$

3.1.2 Pearson Correlation

Definition 2 (Pearson Correlation) *The Pearson correlation coefficient between two (paired) random variables X and Y is defined as*

$$\rho_{X,Y} \doteq \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{3.4}$$

Where cov *is the covariance, and σ_X is the standard deviation of the random variable X, likewise for Y.*

3.1.3 Spearman Correlation

This particular correlation measure has not been used as an evaluation metric for in silico perturbation in the literature, as far as I know, but it has been shown to be more robust than other methods when applied to biological data [14]. It is introduced here since it is a correlation measure, it will then be tested in Part III.

Definition 3 (Spearman Correlation) *The Spearman correlation coefficient between two random variables X and Y is defined as the Pearson correlation between the ranks of the observation in the sample for the variables. Let us first define a rank function R that takes a sample and returns the rank of the instance in the sample, so for example a sample of 2.4, 2.7, 1.5 would become 2, 3, 1*

We can then define Spearman correlation r_s as

$$r_s \doteq \rho_{R(X),R(Y)} = \frac{\operatorname{cov}(R(X), R(Y))}{\sigma_{R(X)} \sigma_{R(Y)}} \tag{3.5}$$

Where cov *is the covariance, and σ_X is the standard deviation of the random variable X, likewise for Y.*

3.2 Methods in the Literature

3.2.1 Wang et al.

The method employed by the authors of **Dictys** [48], presented in Subsection 2.1.1, is the use of Pearson correlation on the logFC of the *average* gene expression levels before and after the perturbation compared to before and after the simulation.

3.2.2 Lotfollahi et al.

The authors of the **CPA** [31] model, presented in Subsection 2.2.1, use an R^2 score on the *average* gene expression levels between the predicted perturbed cells and the ground truth. They also do the same for the top 50 differentially expressed genes (described in Subsection 7.2, they are basically the genes which change more, either increasing or decreasing, therefore more informative for that particular state). Moreover they also compute the R^2 score for the variance of the gene expression levels across the two population compared (of course the variance is intra-population).

3.2.3 Cui et al.

The procedure followed by the authors of the **scGPT** [15] model presented in Section 2.3.1 consists in computing the Pearson correlation on the *average* gene expression levels between the predicted perturbation response and the actual one. They additionally perform the same computation but restricting the genes over which the Pearson correlation is computed to the top 20 differentially expressed genes.

Part II

Personal Contribution

Chapter 4

Benchmarks

I argue that the fact that no two models have the same evaluation procedure is suboptimal for the advancement of the field. For example one of the seminal moments in deep learning was the boost in performance obtained by AlexNet in the ImageNet benchmark [26]. This was definitive proof that deep architectures combined with GPU was the winning paradigm (and as a matter of fact it has been so until today). However the community was convinced of its validity so quickly because they all competed with the same benchmark under the same metric. This work is a first step in trying to replicate this with perturbation prediction models instead of vision ones.

4.1 Components of a Benchmark

The three components of a benchmark are:

- **Dataset**
- **Metric**
- **Evaluation procedure**

Concerning the dataset, I will make use of Adamson & Norman's 2016 perturb-seq experiment [1] that will be described in more detail in Section 11. Of course the choice of dataset is very important in the downstream analysis of the performances that is made. Having a multiplexed perturbation study (i.e. one where couples or triplets of perturbations are applied) is an apt choice, in my opinion, because it allows us to ascertain whether a model is actually learning the intricate interplay between perturbations or if it is only applying the results it “memorized” of any given perturbation. The authors of [2] concur with me on this reasoning, and in fact they employed a similar dataset: Norman 2019 perturb-seq [35].

The selection of the most suitable metric is, of course, the main focus of this work. To select the best among a number of candidate metrics I've devised the spiking protocol described in Section 10.

A word should be spent on the evaluation procedure, which includes choices like the train/test split of the data, among others. It is worth noting that a model can be made to look better than it is by choosing “easy” test data, for example by having it predict unseen cells but from known perturbations, as evidenced in [2].

Chapter 5

Drawbacks of Existing Methods

All the methods mentioned earlier only take into account statistics computed on the marginal distributions, that is the distribution over the expression levels of one gene at a time. As a matter of fact most approaches only look at the average expression of a gene across the cells, which is akin to having bulk data, so much so that in some papers they refer to this processing as pseudo-bulk data.

To better illustrate this point we can look at Figure 5.1 in which a sample of cells is represented: they only have two genes, X and Y. The plot is a heatmap of the expression levels for every cell, and for every gene. The typical approach is to consider the distribution of the read counts across the different genes: we compute the probability that a given piece of RNA we sequenced belongs to a certain class (e.g. gene X), as seen in Figure 5.2. This is basically what's done in bulk and pseudo-bulk data.

A different perspective, that up to now hasn't had much traction in the bioinformatics community is considering the distribution of *cells* in the gene expression levels space. This is what's being illustrated in Figure 5.4. To do so we start from the scatter plot of cells in said space, as shown in Figure 5.3, then we perform Kernel Density Estimation (which will be further explained in Section 8) to obtain the empirical (or estimated) probability distribution.

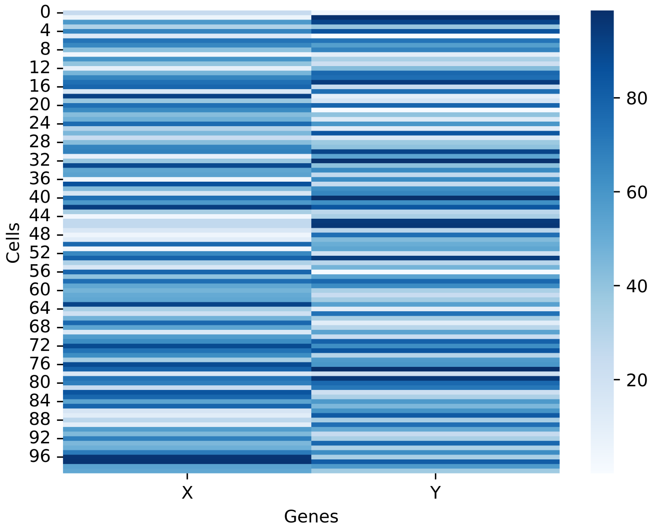

Figure 5.1: Population of Cells

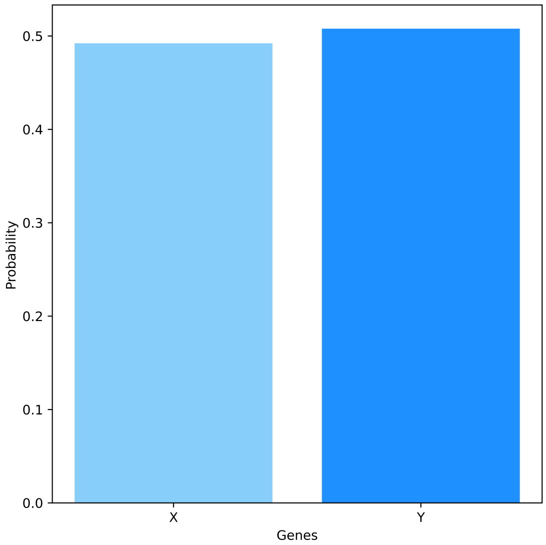

Figure 5.2: Distribution of Read Counts Across Genes

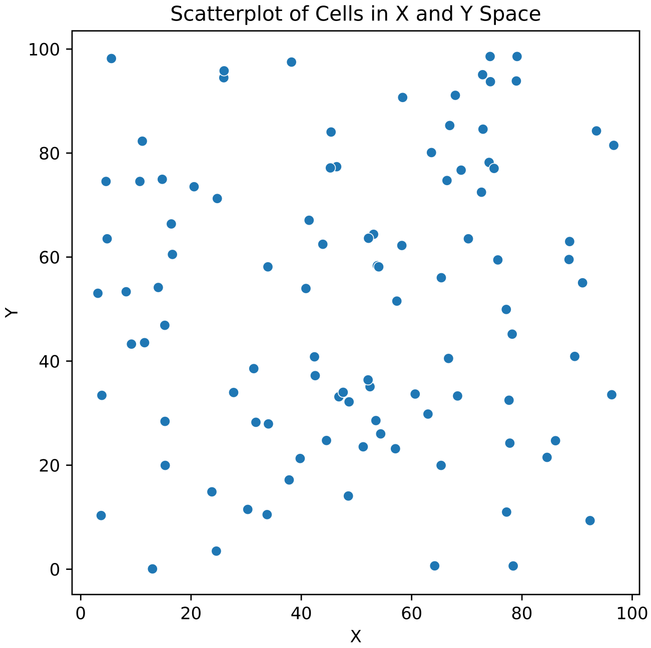

Figure 5.3: Scatter Plot of Cells in the Gene Expression Space

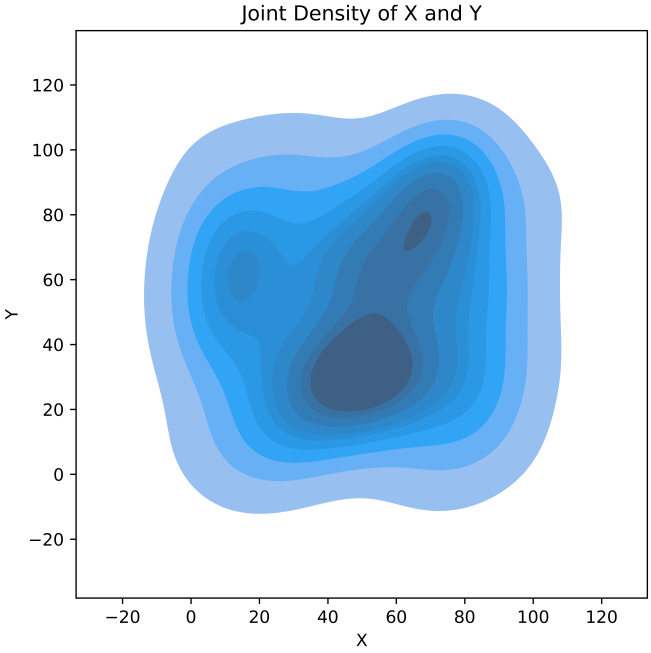

Figure 5.4: Distribution of Cells in the Gene Expression Space (obtained with KDE)

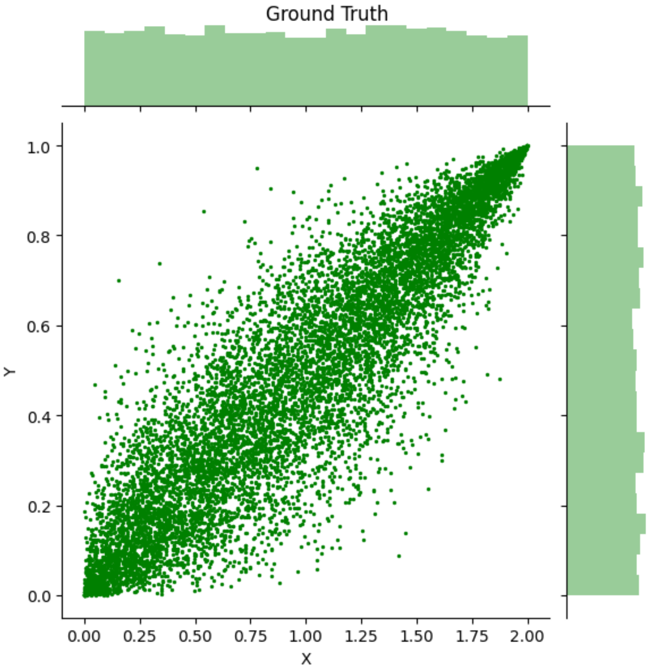

Figure 5.5: Ground Truth

The claim I make with this work is that the way of evaluating models based only on marginals (usually only the average, so even worse) leaves a lot of meat on the table. By which I mean that some models, or more generally some populations, cannot be distinguished through the conventional approach.

Let's take a look at an artificial dataset which illustrates an extreme case in which the usual metrics fail. Take the three populations in Figures 5.5, 5.6, 5.7: the first (Figure 5.5) is the actual result of the perturbation. Then we have two competing models: A (Figure 5.6) and B (Figure 5.7).

It's clear that model B more closely resembles the ground truth, however since all these distributions have the same marginals any approach of the pseudo-bulk kind would be unable to tell them apart and would say that model A is as good as model B.

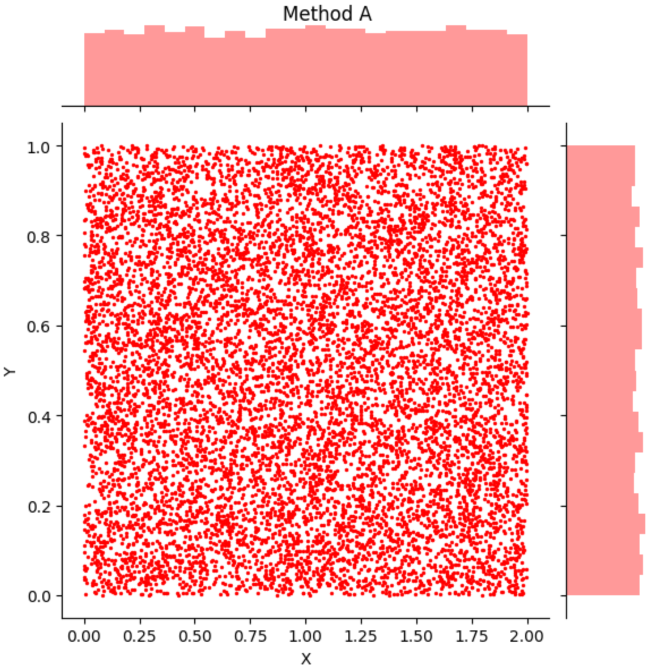

Figure 5.6: Model A

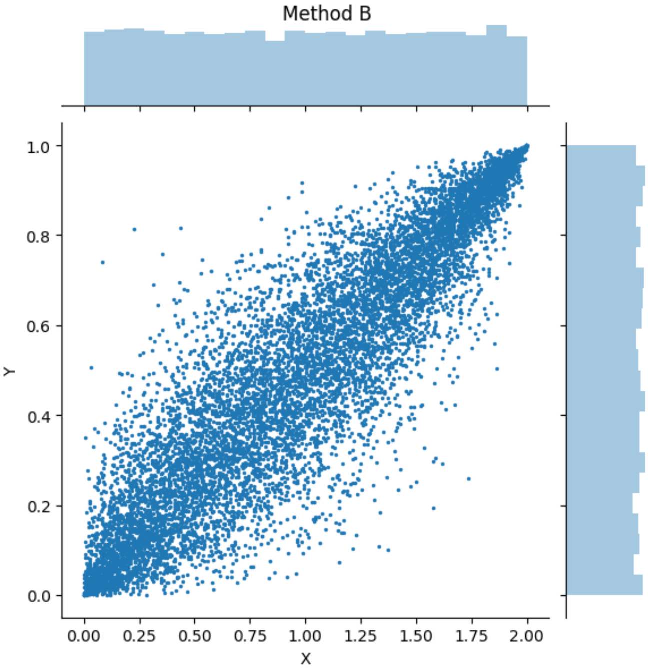

Figure 5.7: Model B

Chapter 6

Proposed Evaluation Method

I argued earlier that using only marginals is not enough to distinguish between some populations when we want to ascertain which is closer to a reference population. Or, in the case of competing models, which among them is most similar to the actual perturbed population.

To take full advantage of the nature of single cell transcriptomic data my proposal is as follows:

- Reduce the dimensionality of the data
- Estimate **joint** probability density function
- Measure the “distance” between distributions

Dimensionality reduction techniques will be discussed in Chapter 7. Then, the estimation method employed will be described in Chapter 8. Lastly, various “distances” between distributions will be presented in Chapter 9.

Let’s work backwards to clarify some points regarding this approach. Our end goal is having a measure that quantifies how similar some populations are, thus allowing to say which among various candidates is the closest to a reference population. We do that on the joint probability distribution in order to maximize the information we are using. Since we want to measure the distance between distributions we first need to obtain an estimate of the population distribution. However there is an impediment to utilizing distances in higher dimensions, it is called the *curse of dimensionality.* This is the reason behind the first step of the procedure.

6.1 Curse of Dimensionality

Let’s take a closer look into the phenomenon known as *curse of dimensionality.*

It is a multi-faceted characteristic of higher dimensional data and affects the effectiveness of various kind of manipulations we can perform on it.

The most noticeable problem is the sparsity of data. If we are given a fixed number of data points and we increase the dimensions we use to describe them it's obvious that the concentration of data points over a "volume" will decrease quite rapidly.

Another issue is the complexity of the algorithms we use on the data, which depend on the dimensionality of it, usually in a linear fashion.

Lastly, and most crucially for our purposes, distances lose meaning in high dimensional spaces. For example it has been shown that under certain assumptions the distance between a point and all other points in a high dimensional dataset tends to the same value as the dimensionality grows [9].

Chapter 7

Dimensionality Reduction

To try to avoid the problems described earlier we will resort to various dimensionality reduction techniques: linear feature extraction in Section 7.1, and three different feature selection techniques quite established in transcriptomics in Sections 7.2, 7.3, 7.4.

7.1 Principal Components Analysis

Principal Components Analysis (PCA) is a linear dimensionality reduction technique. It was invented in 1901 by Karl Pearson [36], maybe the name rings a bell (interestingly he did not invent the Pearson correlation though, it was Auguste Bravais [12]).

It can be viewed in different ways, for example as the best fitting ellipsoid [28] to our data points. Alternatively we can view PCA as the iterated procedure of finding orthogonal directions that minimize the reconstruction error if we project data onto the subspace they define. This ties in with Eckart-Young theorem [17] and in fact we can also see PCA as equivalent to performing a singular value decomposition (SVD) on the data matrix.

7.2 Differentially Expressed Genes

As stated in [23], differential gene expression analysis “attempts to infer genes that are statistically significantly over- or under-expressed between any compared groups (commonly between healthy and condition per cell type)”.

There are various approaches to perform it but they all come down to quantifying the magnitude of the change (either positive or negative) in a given gene’s expression. Also the significance of said variation is to be estimated. By jointly considering these two quantities we can recover the most differentially expressed genes. We will use the routines in [52], using them in the way described in [31], in order to make the results as close as possible to the reference method.

Selecting the 50 (as we will do) or 20 most differentially expressed genes (DEG) is a very decisive way of performing feature selection considering the starting point of circa 20,000 genes: it is a reduction of 2 or 3 orders of magnitude in the number of features.

7.3 Highly Expressed Genes

Another feature selection technique quite often used in systems biology is the choice of highly expressed genes (HEG). This is the methodology employed in [2] for example.

To obtain the most highly expressed genes we first normalize the counts for every cell, so as to make every cell contribute equally to our considerations. Then we compute the average value of the normalized expression across the different cells, for every gene. We then rank them according to their average normalized expression levels and take the top k, where k is typically a thousand.

7.4 Highly Variable Genes

The selection of highly variable genes (HVG) is a technique first introduced in [41]. There are different flavours but we will stick with the original one in this work.

We start by calculating the average expression levels of the genes and their dispersion (coefficient of variation). Then we put the genes in 20 bins according to their average expression levels. Within each bin we compute the z-score for the dispersion measure (which is the coefficient of variation). This allows us to identify, within each bins, the genes that varied the most. Both computing the coefficient of variation instead of the standard deviation, and also performing the selection within bins of similarly expressed genes, help in addressing the heteroscedasticity of the genes' expression distribution.

Chapter 8

Kernel Density Estimation

Kernel density estimation, or KDE for short, is a statistical technique used to obtain an estimate of a continuous probability density from a finite amount of samples [44].

Definition 4 (Kernel Density Estimate) *Given a non-negative kernel function K, a bandwidth parameter $h > 0$, and a set of points x_i a kernel density estimator $\hat{f}$ is defined as*

$$\hat{f}_h(x) \doteq \frac{1}{n} \sum_i K_h(x - x_i) = \frac{1}{hn} \sum_i K\left(\frac{x - x_i}{h}\right) \tag{8.1}$$

The bandwidth parameter determines how close the kernel density estimate is to the actual data in quite a drastic manner, even more so than the choice of kernel function (which will be gaussian in this work).

To choose the best value of the bandwidth we will resort the so called Scott's rule [43], and so we will set the bandwidth h to be

$$h = n^{-\frac{1}{d+4}}$$

where d is the number of dimensions of the data and n is the amount of data points. This is by no means assured to be optimal, but there's no agreed upon "best" method in the literature.

Chapter 9

Quantifying Similarity Between Distributions

9.1 Kullback-Leibler Divergence

Kullback-Leibler divergence, or *relative entropy*, is an information theoretical quantity that can be used to measure the distance between two distributions, first described in [29].

Let us take a step back and introduce some related concepts first in order to give a better intuitive understanding of what Kullback-Leibler (KL) divergence is. First we will define some quantities only considering a single distribution, then we will adapt these concepts to confront different ones. We will consider discrete random variables at first, then extend the concepts to the continuous case.

9.1.1 Surprisal

We first try to define the information contained in an event E. We call this quantity *self-information*, or *surprisal*, $S(E)$.

We would like it to have the following properties:

- $S(E)$ is monotonically decreasing with the probability of E. Meaning that the more E is likely, the less information it can possess.

- $S(E) = 0$ if, and only if, the probability associated with E is one. That is to say, if an event always occurs then it cannot communicate any information.

- $S(AB) = S(A) + S(B)$ or, the information of two independent events A and B occurring together is the sum of the information of the event A and event B.

These desiderata are arguably quite reasonable, and an allow us to define an actionable information quantity. They also align quite closely to what we would intuitively call the surprisal, or how much we would be surprised by an event occurring, given that we know its chance of happening.

Definition 5 (Surprisal) *Let E be an event and p(E) the probability of said event, we define a surprisal quantity S(E) associated with that event occurring as*

$$S(E) \doteq \log\left(\frac{1}{p(E)}\right) \tag{9.1}$$

It's easy to see that this satisfies our desiderata, as a matter of fact it can be shown that the family of functions given by the logarithm of the inverse of the probability of the event, with as parameter the base of the logarithm, is the only family of functions that satisfies our desiderata.

9.1.2 Entropy

Now we use the concept of *surprisal* we defined above and apply it to a random variable. Namely we take its expected value, which is roughly the average of the surprisal across all the events in the sample space, weighted by their chance of occurring.

Definition 6 (Shannon Entropy) *Let X be a random variable, we define the Shannon Entropy associated with it as the expected value of the surprisal of the events from its sample space.*

$$H(X) \doteq \mathbb{E}[S(X)] \tag{9.2}$$

9.1.3 Cross-Entropy

Now we finally introduce a second distribution, we basically compute the Shannon entropy of distribution Q but using the expected value under distribution P's probabilities. More formally

Definition 7 (Cross Entropy) *Let $p(x)$ and $q(x)$ be two probability density functions associated with the random variables P and Q respectively, we define the cross-entropy H(P, Q) as*

$$H(P,Q) \doteq \mathbb{E}_P[S(Q)] = \sum_x p(x)\log\left(\frac{1}{q(x)}\right) \tag{9.3}$$

This quantity tells us how surprised we will be by employing distribution Q as a model for distribution P. The drawback is that it does not take into account how surprised we would be if we used P as a model for distribution P, this last quantity being the upper limit for the goodness of a model (i.e. lower limit on expected surprisal) of course, and corresponding to the concept of entropy. We are now ready to introduce the Kullback-Leibler divergence.

9.1.4 KL Divergence

Definition 8 (Kullback-Leibler Divergence) *Let $p(x)$ and $q(x)$ be two probability density functions associated with the random variables P and Q respectively, we define the KL divergence between P and Q (order matters)* $\mathrm{D_{KL}}(P||Q)$ *to be*

$$\mathrm{D_{KL}}(P||Q) \doteq \int_{x\in\mathbb{R}^d} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx \tag{9.4}$$

It is easy to see that this quantity is equivalent to the cross-entropy between P and Q minus the entropy of distribution P.

9.2 Jensen-Renyi Divergence

This divergence is closely related to the more established Jensen-Shannon divergence, it is basically its equivalent but using Renyi entropy instead of Shannon entropy. There exists a more general version for an arbitrary number of distributions but we will limit ourselves to the case where we confront only two.

Definition 9 (Jensen-Renyi Divergence) *Let P and Q be two random variables, H_α be the Renyi entropy of order α and π_P and π_Q two postive numbers summing up to one, then the Jensen-Renyi divergence* $\mathrm{JR}_{\pi,\alpha}(P,Q)$ *is defined as*

$$\mathrm{JR}_{\pi,\alpha}(P,Q) \doteq \mathrm{H}_\alpha(\pi_P P + \pi_Q Q) - \pi_P \mathrm{H}_\alpha(P) - \pi_Q \mathrm{H}_\alpha(Q) \tag{9.5}$$

9.2.1 Renyi Entropy

Renyi entropy can be seen as a generalization of Shannon entropy.

Definition 10 (Renyi Entropy) *Given a random variable X and for $0 < \alpha < \infty$ and $\alpha \neq 1$ it is defined as follows*

$$\mathrm{H}_\alpha(X) \doteq \frac{1}{1-\alpha}\log\int_{x\in\mathbb{R}^d} f^\alpha(x)\,dx \tag{9.6}$$

It can be shown that the limit of for $\alpha \to 1$ of Renyi entropy is Shannon entropy [13].

9.2.2 Closed Form for Mixtures of Gaussians

We will restrict the derivations for the case where all the centroids' associated gaussian have equal variance-covariance matrix (of the form $\sigma^2\mathbf{I}$), and Jensen-Renyi is such that $\pi_P = \pi_Q = \frac{1}{2}$. The same techniques shown here can be used to obtain more general formulas but since they will not be employed in the rest of this work the ones presented will suffice. The calculations closely follow the procedure employed in [47].

Let V be a *gaussian mixture* such that

$$f_V(x) = \sum_i \omega_i G(x - v_i, \sigma^2\mathbf{I}) \tag{9.7}$$

where G is the standard multivariate gaussian probability density function. Since we want the gaussian mixture to be a probability density function it's clear that the ω_i will be such that $\sum_i \omega_i = 1$.

We can see that the Renyi Entropy of order 2 has closed form for such a distribution.

$$\begin{gathered}\mathrm{H}_2(V) = -\log \int_{\mathbb{R}^d} f_V(x)^2 \, dx = \\ -\log \sum_i \sum_j \int_{\mathbb{R}^d} \omega_i \omega_j G(x - v_i, \sigma^2\mathbf{I}) G(x - v_j, \sigma^2\mathbf{I}) \, dx = \\ -\log \sum_i \sum_j \omega_i \omega_j G(v_i - v_j, 2\sigma^2\mathbf{I})\end{gathered} \tag{9.8}$$

Where we swapped the summations with the integral operations between the first and second line and performed square completion between the probability density functions of the gaussian distributions to go from the second line to the end result.

Now if we consider that a convex combination of probability density functions which are mixture of gaussians is itself a probability density function (since it's a convex combination) of a mixture of gaussians (since it's the sum of gaussians pdfs) then we can easily obtain a closed form for the Jensen-Renyi divergence.

$$\begin{aligned}\mathrm{JR}_{\pi,2}(P,Q) = -\log \frac{1}{M^2} \sum_i^M \sum_j^M G(u_i - u_j, 2\sigma^2\mathbf{I}) \\ + \frac{K_P}{M} \log \frac{1}{K_P^2} \sum_i^{K_P} \sum_j^{K_P} G(v_i - v_j, 2\sigma^2\mathbf{I}) \\ + \frac{K_Q}{M} \log \frac{1}{K_Q^2} \sum_i^{K_Q} \sum_j^{K_Q} G(w_i - w_j, 2\sigma^2\mathbf{I})\end{aligned} \tag{9.9}$$

Where K_P and K_Q are the number of gaussians in P and in Q respectively and $M = K_P + K_Q$. The v_i and w_i are respectively the centroids of P or Q and u_i are the centroids of the distribution obtained by averaging P and Q.

9.3 Maximum Mean Discrepancy

9.3.1 Reproducing Kernel Hilbert Space

A reproducing kernel Hilbert space (RKHS) is an **Hilbert space** on which we can define a **kernel function** that satisfies Mercer's condition, and also possesses the **reproducing** property [27].

We will now untangle this definition and clarify every component.

An Hilbert space is a vector space that may be infinite-dimensional and is equipped with an inner product. It is also a complete metric space if the inner product is considered as a distance.

A kernel function is a positive semidefinite function of two points that returns a real value in our case.

The reproducing property says that the pointwise evaluation of a function, that belongs to the Hilbert space we are referring to, can be done by taking the inner product of the reproducing kernel where one of its arguments is fixed to the point we want to evaluate the function in.

An RKHS therefore enables us to compute the value of a function at various points without reconstructing the function itself. It comes in especially handy since it allows us to avoid estimating the probability density of the data.

9.3.2 MMD

Maximum mean discrepancy (MMD), introduced in [21]) is an approach to compute a distance between distributions by leveraging the qualities of a RKHS.

Definition 11 (MMD) *Given two populations X and Y with samples drawn from them respectively x_i, of cardinality m, and y_i, of cardinality n we can define the maximum mean discrepancy between them, according to kernel function k, as*

$$\begin{aligned} \mathrm{MMD}_k(X,Y) \doteq \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m k(x_i, x_j) + \frac{1}{n(n-1)} \sum_i^n \sum_{j \neq i}^n k(y_i, y_j) \\ - \frac{2}{mn} \sum_i^m \sum_j^n k(x_i, y_j) \end{aligned} \tag{9.10}$$

A thing worth mentioning is the very similar functional form of the MMD distance compared to Jensen-Renyi divergence.

9.4 Wasserstein Distance

Wasserstein distance, also known as Earth Mover's distance, is a metric that leverages concepts from optimal transport theory. Unlike other "distances" I've introduced before, this one is an actual metric (in the mathematical sense), i.e. it satisfies the triangle inequality, is symmetric, and is zero only if the objects confronted are the same.

The distance is obtained by solving an optimization problem. The objective to be minimized is the total cost of a transport plan, under the constraint that said transport plan is such that it morphs the first distribution into the second.

A transport plan is basically a mapping that tells us how much probability has to be moved from any given category (in the case of discrete distributions) to any other in order to obtain the second distribution by starting from the first.

It is called Earth Mover's distance because the first probability distribution can be interpreted as a set of mounds of earth, and the second as a set of holes, and thus the transport plan can be interpreted as the movements of earth from the mounds to the holes necessary to obtain a level field.

A thorough discussion on this distance can be found at [37] along with the discrete and continuous proper formulations.

9.4.1 Sinkhorn Distance

Sinkhorn distance is a version of Wasserstein distance where the objective function to be minimized is not only the cost of the transport plan, but it also has an entropic regularization term, that is, a penalty for having too complex a plan. It was introduced in [4] and its main advantage is that we can compute it in near-linear time, as opposed to the quadratic time necessary for regular Wasserstein distance. Moreover Wasserstein distance is the limit case of Sinkhorn as we diminish the regularization term in the objective function, so we can obtain quite similar results.

Part III

Experimental Results

Chapter 10

Study Design

Now we are to answer quite a thorny question: how do we evaluate a method of evaluation? It seems like a case of a dog eating his own tail.

If, under a useful metric, a simulation is very close to the actual perturbation, then we can say that the model used to simulate it is good. When we compare various models the one closer to the actual perturbation is considered to be the best, of course. How can we say that a metric is useful? We need a way to evaluate metrics.

To break this impasse we have to make some assumptions, that hopefully are justified. We will assume that the more a metric is able to tell apart distributions that are similar (but not the same), the more useful it is. This seems like a reasonable assumption, and assuming the opposite would be clearly nonsense. We have already mentioned a case where a metric is not able to tell apart distributions in Section 5.

However the example could be construed as unnatural, or not representative of what happens in actual biological data. We will then make another assumption: that the *spiking protocol* that creates similar distributions (that I will present shortly) is representative of different perturbations occurring in cell populations.

10.1 Spiking Protocol

The spiking protocol is as follows: we start from two populations A and B. These populations are perturbed states of the same cell line. Then we draw a sample from population A that we will call “sub A”, composed exclusively by cells belonging to population A. We also create sample “ref A” in much the same manner. To create the “spiked” sample instead we take a certain percentage from population B and another percentage from population A, this last percentage will be called *spike* percentage. A **crucial** point is any given cell belongs to either one of the

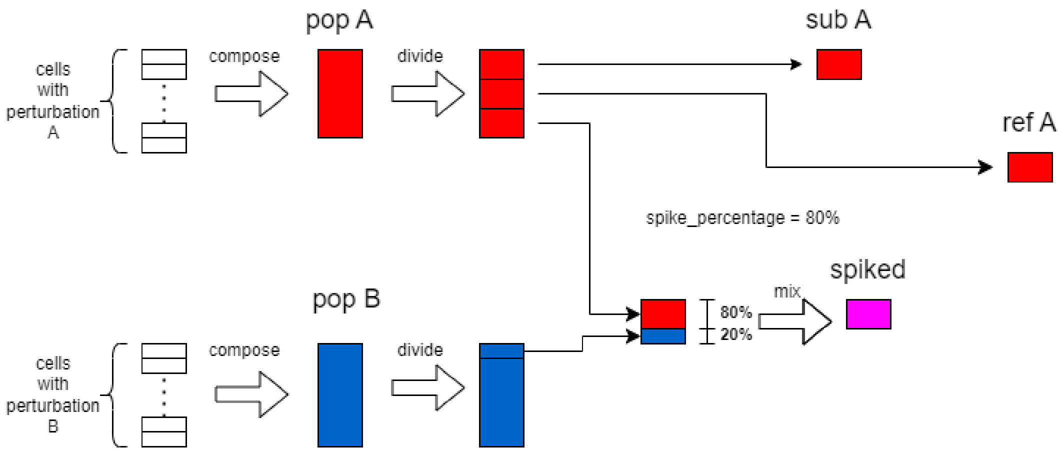

Figure 10.1: Diagram of the spiking protocol

three samples, that is to say the samples have no cell in common. Also, all the samples have the same *exact* number of cells.

At spike percentage 0% the various samples will be just subsamples of the respective populations A (for sub A and ref A) and B (for the spiked sample). At spike percentage 100% the three samples will all be subsamples (with no overlap though!) drawn from population A.

A diagram can be seen in Figure 10.1. The vertical bars represent the cells belonging to the various populations. In the case of the image we can see that the spike percentage is 80% for example.

Once we have sub A and the spiked sample we confront them with the reference sample raf A, let's say that the metric is called τ, then we compute $\tau(\text{subA}, \text{refA})$ and $\tau(\text{spiked}, \text{refA})$. We expect the distance between sub A and the reference sample ref A to be less than the distance between the spiked sample and the reference. This is by construction, so we have good reason to believe it to be true.

This allows us to assign an accuracy score to a metric by repeating this process with different samples. Of course as the spike percentage increases we expect any metric to become less able to tell the samples apart, and, in the extreme case of spike percentage 100%, we expect the accuracy to be around 50%.

We can see a plot of the accuracy as the spike percentage increases (average across all perturbation couples) with 95% confidence bands in Figure 10.2. We see that, as expected, as the spike percentage goes to 100% the accuracy drops to 50%.

We will focus on the accuracy when the spike percentage is 80% in the following. This allows us to perform more repeated tries and obtain higher statistical power of the hypothesis tests we will employ. We will also have a single number on

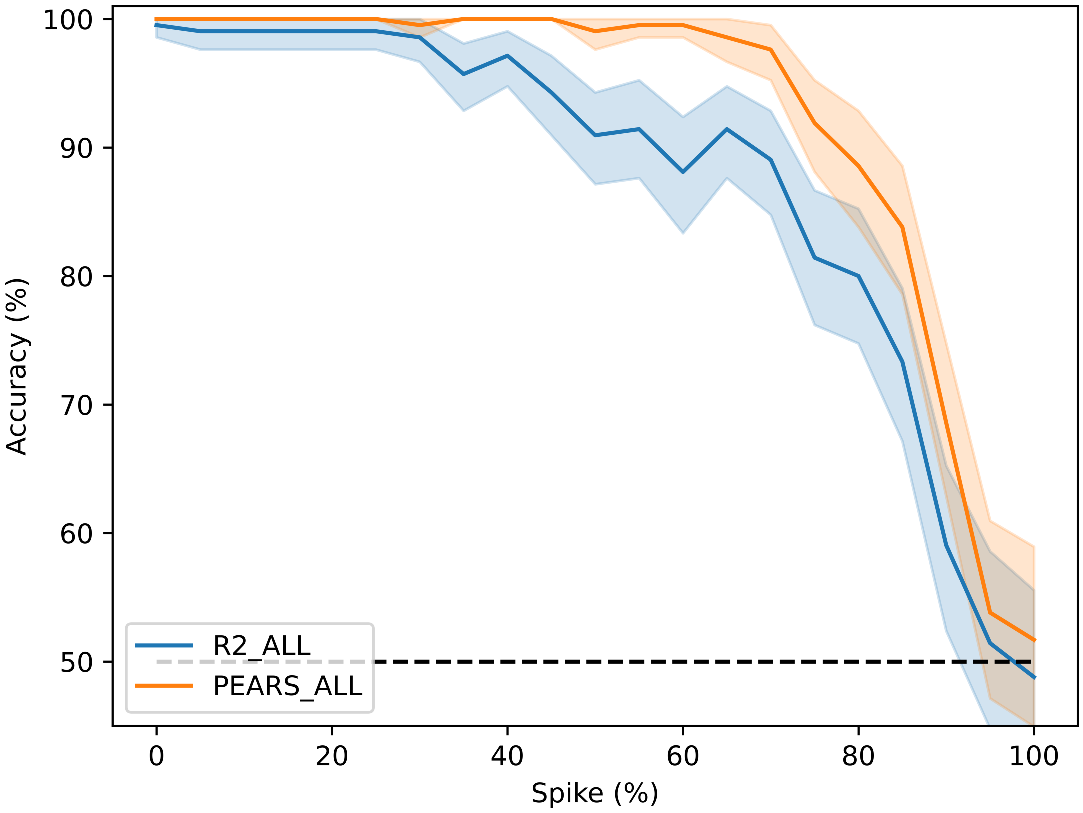

Figure 10.2: Plot of the accuracy of various metrics

which to perform hypothesis testing, which is more manageable than multiple data points per run. I believe that this choice is warranted by the shape of these curves we see in this example and from the considerations that there must be a fixed point at 100% spike and 50% accuracy for every metric (also it seems that at 0% spike percentage the metrics reach almost 100% accuracy), moreover it is not unreasonable to assume these curves to be monotonically non-increasing (and in fact they seem to be so), which renders more sensible to test their behaviour only at one point close to the fixed point.

Chapter 11

Dataset

The dataset I've elected to use is Adamson & Norman's 2016 perturb-seq experiment [1]. As I've already mentioned in Section 4.1 this dataset is particularly suited for our purposes since it is a multiplexed perturbation study: it has single perturbations, couples of those perturbations, and also triplets. This enables us to see whether the models are learning how the perturbations interact, beyond the additive effect we would obtain with a linear model.

The identifier Gene Expression Omnibus [18] [8] code for the dataset is **GSE90546**. More specifically we will only use the samples coming from the epistasis experiment, with identifier **GSM2406677**.

The perturbation in this subset of the Adamson & Norman's dataset are the following:

- Single Perturbations
  - `3x_neg_ctrl_pMJ144-1`
  - `3x_neg_ctrl_pMJ144-2`
  - `ATF6_only_pMJ145`
  - `IRE1_only_pMJ148`
  - `PERK_only_pMJ146`
- Double Perturbations
  - `ATF6_IRE1_pMJ152`
  - `ATF6_PERK_pMJ150`
  - `PERK_IRE1_pMJ154`
- Triple Perturbations
  - `ATF6_PERK_IRE1_pMJ158`

Chapter 12

Results

Here we will confront different metrics' accuracies at 80% spike. The accuracy refers to the amount of times the metric respects the following inequality

$$\tau(\text{subA}, \text{refA}) < \tau(\text{subA}, \text{spiked})$$

We've mentioned in Section 10.1 how it makes sense to assume subA and refA to be "closer" than subA and the spiked sample. For this to be true the various correlation metrics are made to resemble a distance by taking $\tau = 1 - \rho$ where ρ is the correlation.

We will randomly draw a pair of perturbation and confront the various metrics on the same samples, in order to have a *ceteris paribus* condition and therefore obtain some **paired statistics**. The procedure is repeated one thousand times.

Moreover, for every pair of metrics, we will test the hypothesis that the first is worst than the second using Wilcoxon's signed-rank test on the difference of the paired statistics [51].

We basically test whether the distribution given by the difference of the "correctness" (i.e. 1 if correct and 0 if not, for every random draw and for every metric) is symmetric around the origin. The test will of course be one-sided, so the null hypothesis is that the metrics are equal or the second is superior to the first, and the alternative is that the first is worse than the second (i.e. the tested distribution's average is greater than zero).

12.1 Marginal Metrics

Let us start with the metrics that work only on marginals.

We obtain the average accuracy of the metrics, displayed in Table 12.1, from the repeated paired statistics.

The uncorrected results of the Wilcoxon test are shown in Figure 12.1.

Method	ALL	DEG_50	HEG_1000	HVG_1000	PCA_7
COS	86.49%	87.41%	86.02%	90.27%	64.32%
L2	74.36%	83.17%	75.98%	86.18%	72.43%
PEARS	86.72%	86.87%	85.17%	**90.42%**	62.70%
R2	74.29%	83.24%	76.06%	86.10%	72.36%
SPEAR	51.20%	75.60%	74.05%	55.29%	58.84%

Table 12.1: Accuracies of marginal metrics

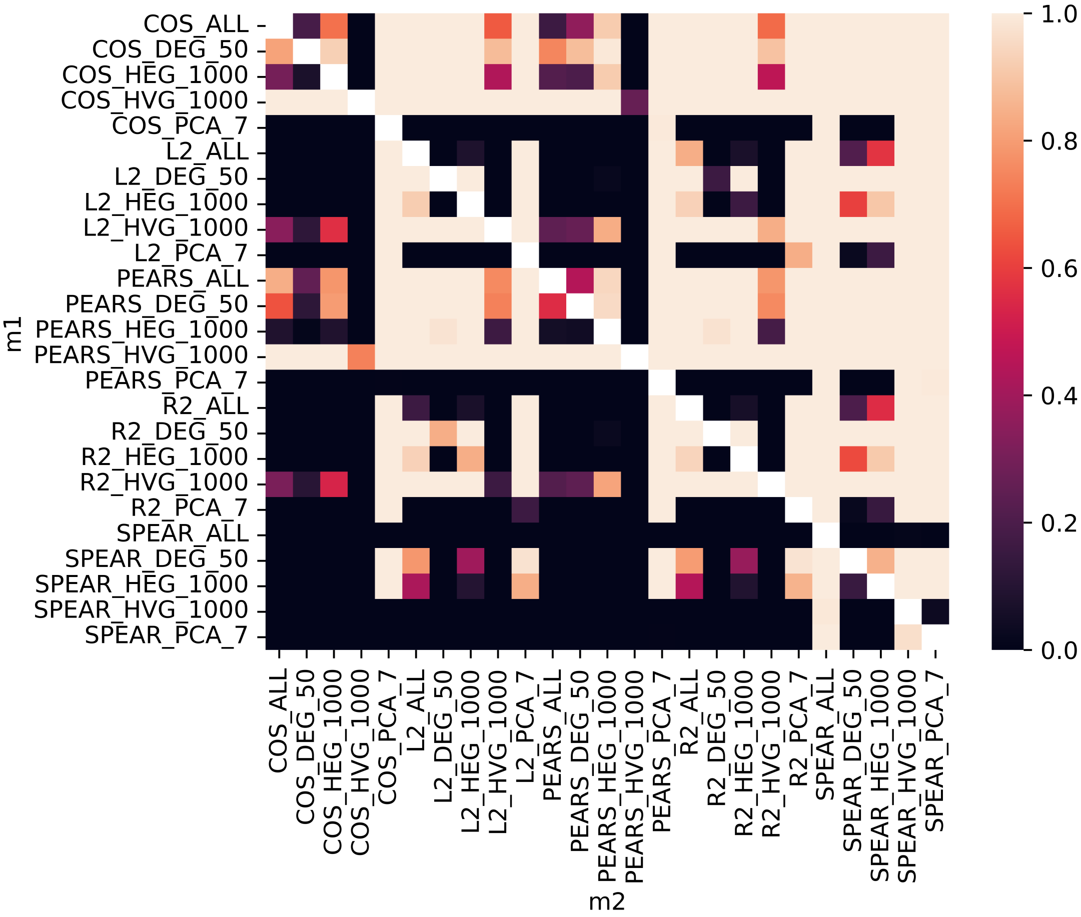

Figure 12.1: Uncorrected Wilcoxon test results

Method	ALL	DEG_50	HEG_1000	HVG_1000	PCA_7
JR	-	49.71%	-	-	53.57%
KL	-	49.81%	-	-	83.20%
MMD	51.93%	72.10%	51.83%	57.92%	52.12%
NRG	76.54%	86.87%	76.93%	88.99%	79.25%
PEARS	89.58%	88.61%	91.70%	**91.99%**	64.86%
R2	76.16%	85.33%	73.75%	88.22%	73.55%

Table 12.2: Accuracies of marginal metrics

However, since we are performing multiple comparisons, we know these p-values will not reflect the actual probability of rejecting the null hypothesis when it is true on all the comparisons simultaneously. We therefore resort to Bonferroni's correction [11]. The results of the test after applying it are reported in Figure 12.2, where I've masked the non-significant ones at a 5% overall significance.

We can see that (except for Spearman correlation) the most effective preprocessing procedure was the selction of highly variable genes, followed by differentially expressed genes, then highly expressed genes and lastly PCA. The exception given by Spearman makes sense if we consider how exactly the highly variable genes are extracted, since we are "forcing" some lowly expressed genes to be in the top thousand chosen (by means of the bins) in those genes it's likely that rank has little to no significance, and therefore Spearman's correlation is very noisy with this preprocessing procedure.

Concerning the metrics instead we clearly see that Pearson and Cosine similarity behave very similarly, as well as R2 and L2. This is by no means a surprise given their very functionally similar form. The clear winners of this contest are Pearson and Cosine similarity.

12.2 Joint Distribution Metrics

We now run the same procedure with two representatives of the marginal metrics: R2 and Pearson. We compare them with the metric that work in the space of genes' expression levels on the joint probability distribution: Kullback-Leibler divergence (KL), Jensen-Renyi divergence (JR), maximum mean discrepancy with gaussian kernel (MMD), energy distance (NRG) and Sinkhorn distance (SINK). The accuracies are reported in Table 12.2. The uncorrected Wilcoxon tests are in Figure 12.3 and the corrected ones are in Figure 12.4.

As we can see none of the metrics that work on the joint probability distribution are better than Pearson correlation. This could be due to the difficulty in estimating the probability density from so few data points in such a high-dimensional

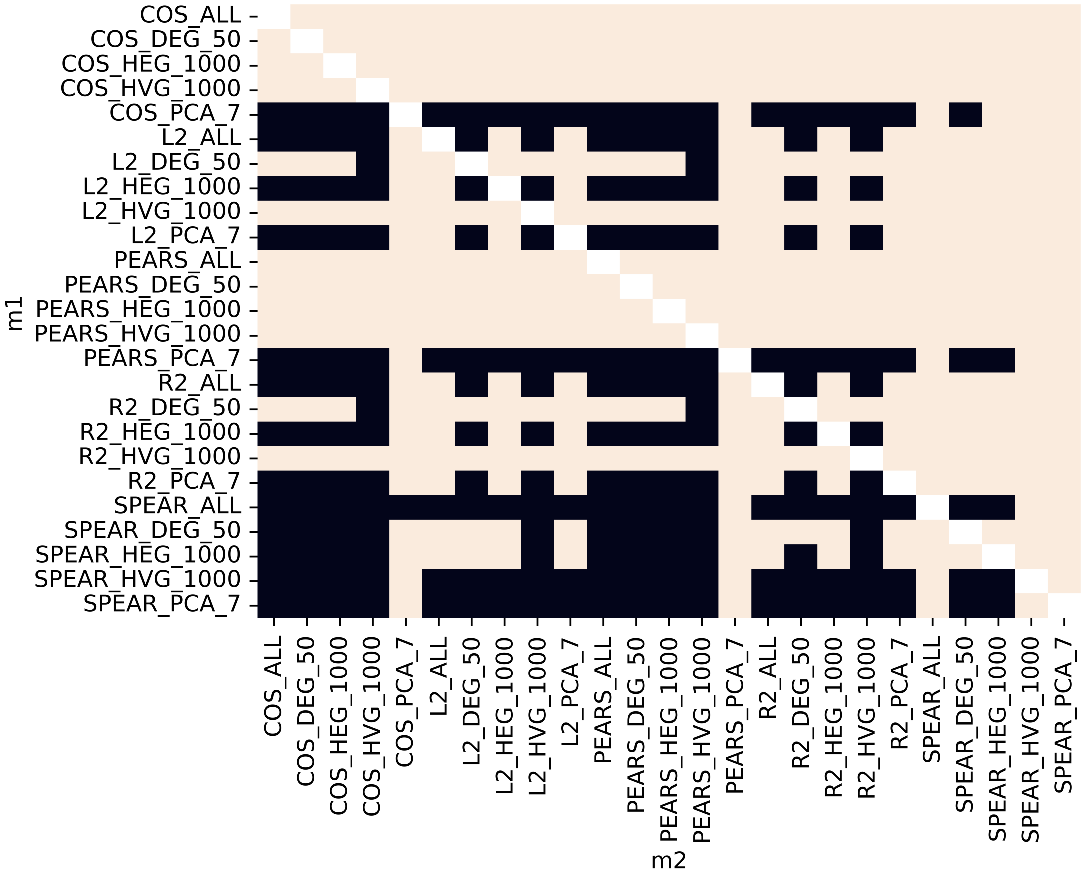

Figure 12.2: Bonferroni corrected and masked Wilcoxon test results, dark means significant

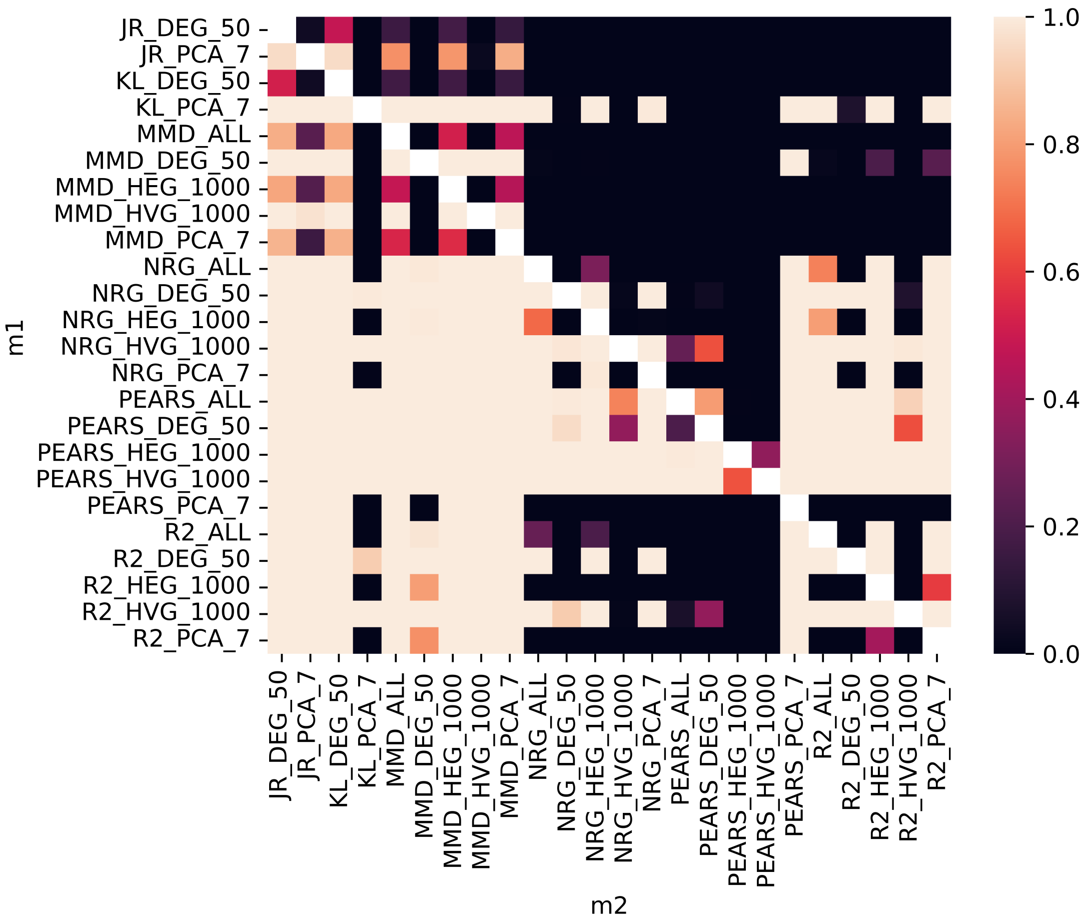

Figure 12.3: Uncorrected Wilcoxon test results for joint distribution metrics

spaces. The most effective seems to be the Energy distance, followed by maximum mean discrepancy and the least effective is Jensen-Renyi divergence.

I argue that this could be due to the fact that the Energy distance is computed on a simple norm, whereas MMD is a squared norm and JR on a squared norm under a logarithm. My hypothesis is that this exacerbates the curse of dimensionality problem I've mentioned in Section 6.1.

The Kullback-Leibler divergence instead is even more peculiar: it is the worst performing one with the top 50 differentially expressed genes (tied with JR divergence) but it is also the best performing one when we apply PCA as the preprocessing step. This, again, could be due to the curse of dimensionality problem and the scarce number of the data points.

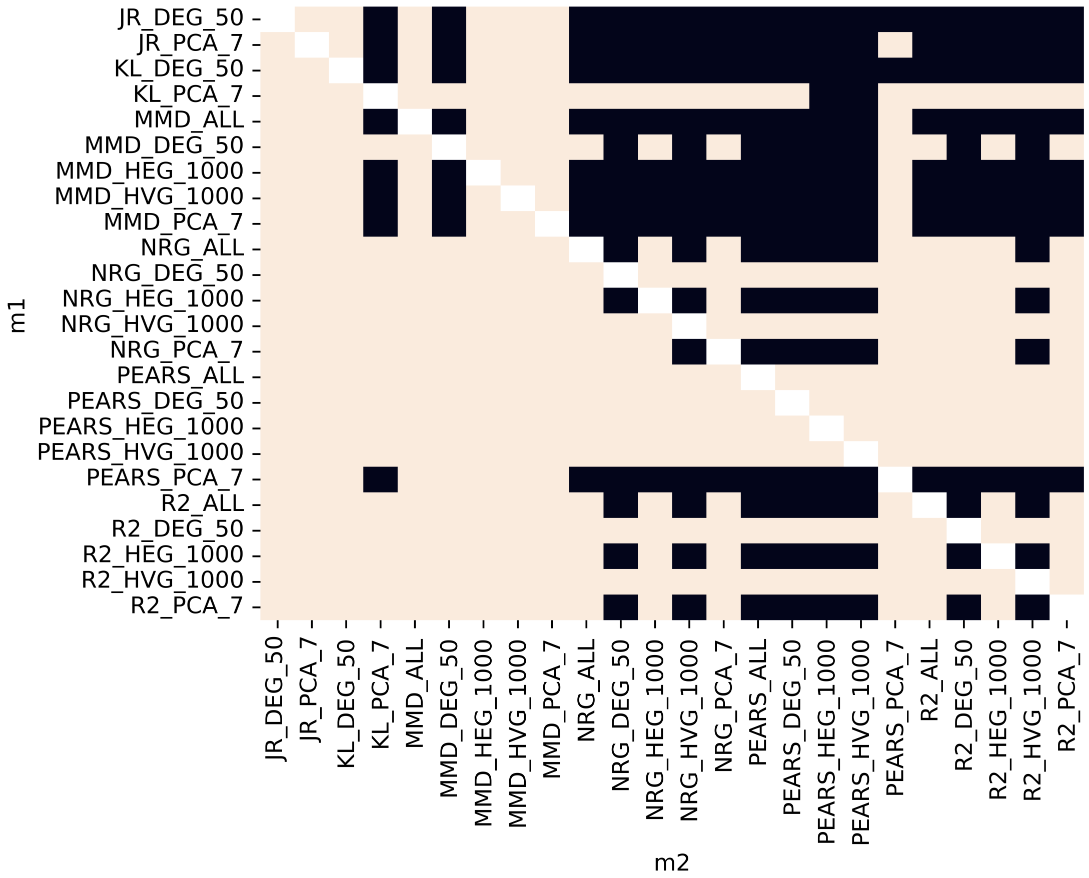

Figure 12.4: Bonferroni corrected and masked Wilcoxon test results, dark means significant

Chapter 13

Discussion

Let us summarize the contents of this manuscript. First I briefly gave a primer on the scientific endeavour of understanding the intricate genetic machinery inside the cell. I also gave a cursory view on the need and utility of in silico perturbation methods. Then I argued for the necessity of an agreed upon benchmark on which to have the models compete, which, as evidenced by a couple of very recent publications [53] [2], is something that is being acknowledged by the community.

Then I set out to answer the question of how to pick the best metric, and I proposed the spiking protocol which is, in my opinion, a satisfactory answer. Through the extensive experimentations performed on the various metrics and preprocessing methods the clear winners were: Cosine similarity, or Pearson correlation, as the metric, and selection of the top 1,000 most highly variable genes as the dimensionality reduction technique, among the ones tested of course.

Lastly, I highlighted some criticalities of pseudo-bulk evaluation and tried to come up with a method that takes full advantage of the nature of single cell data. All of the novel methods I proposed relied on estimating the probability density of the single cells, either directly or through the kernel trick. Sadly, none of them worked.

It seems like the problem is the sparsity of cells in the high dimensional spaces thus making the probability distribution estimation hard. Therefore a possibility worth mentioning could be to map the single cells (with optimal transport) to a corresponding perturbed equivalent. This could circumvent the problem of the lack of before and after of the same cell, by having a *virtual* “after” for every “before” cell.

It is maybe a bit premature to go beyond pseudo-bulk evaluation right now, also considering that the current models cannot manage to saturate the existing metrics, as evidenced by the paper mentioned above. In any case, I maintain the validity of this avenue of inquiry for the future and I hope to be able to continue this work.

Bibliography

[1] Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, Jacqueline E. Villalta, Luke A. Gilbert, Max A. Horlbeck, Marco Y. Hein, Ryan A. Pak, Andrew N. Gray, Carol A. Gross, Atray Dixit, Oren Parnas, Aviv Regev, and Jonathan S. Weissman. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882.e21, December 2016.

[2] Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear methods. September 2024.

[3] Bruce Alberts, Dennis Bray, Julian Lewis, Martin Raff, Keith Roberts, James D Watson, et al. *Molecular biology of the cell*, volume 3. Garland New York, 1994.

[4] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration, 2018.

[5] Pau Badia-i Mompel, Lorna Wessels, Sophia Müller-Dott, Rémi Trimbour, Ricardo O. Ramirez Flores, Ricard Argelaguet, and Julio Saez-Rodriguez. Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews Genetics*, 24(11):739–754, June 2023.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[7] Lucy W. Barrett, Sue Fletcher, and Steve D. Wilton. Regulation of eukaryotic gene expression by the untranslated gene regions and other non-coding elements. *Cellular and Molecular Life Sciences*, 69(21):3613–3634, April 2012.

[8] Tanya Barrett, Stephen E. Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F. Kim, Maxim Tomashevsky, Kimberly A. Marshall, Katherine H. Phillippy, Patti M. Sherman, Michelle Holko, Andrey Yefanov, Hyeseung Lee, Naigong Zhang, Cynthia L. Robertson, Nadezhda Serova, Sean Davis,

and Alexandra Soboleva. Ncbi geo: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995, November 2012.

[9] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. *When Is "Nearest Neighbor" Meaningful?*, page 217–235. Springer Berlin Heidelberg, 1999.

[10] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[11] Carlo Emilio Bonferroni. *Teoria statistica delle classi e calcolo delle probabilità*. Pubblicazioni del R. Istituto superiore di scienze economiche e commerciali di Firenze 8. Seeber, Firenze, 1936.

[12] Auguste Bravais. *Analyse Matheematique Sur Les Probabilites Des Erreurs de Situation d'Un Point*. Impr. Royale, 1844.

[13] P.A. Bromiley, N.A. Thacker, and E. Bouhova-Thacker. *Shannon Entropy, Rényi Entropy, and Information*. 2004.

[14] F M Buffa, A L Harris, C M West, and C J Miller. Large meta-analysis of multiple cancers reveals a common, compact and highly prognostic hypoxia metagene. *British Journal of Cancer*, 102(2):428–435, January 2010.

[15] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[17] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936.

[18] R. Edgar. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, January 2002.

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[20] Boris Görke and Jörg Stülke. Carbon catabolite repression in bacteria: many ways to make the most out of nutrients. *Nature Reviews Microbiology*, 6(8):613–624, August 2008.

[21] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[22] Matthias Heidenreich and Feng Zhang. Applications of crispr–cas systems in neuroscience. *Nature Reviews Neuroscience*, 17(1):36–44, December 2015.

[23] Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Hananeh Aliee, Meshal Ansari, Pau Badia-i Mompel, Maren Büttner, Emma Dann, Daniel Dimitrov, Leander Dony, Amit Frishberg, Dongze He, Soroor Hediyeh-zadeh, Leon Hetzel, Ignacio L. Ibarra, Matthew G. Jones, Mohammad Lotfollahi, Laura D. Martens, Christian L. Müller, Mor Nitzan, Johannes Ostner, Giovanni Palla, Rob Patro, Zoe Piran, Ciro Ramírez-Suástegui, Julio Saez-Rodriguez, Hirak Sarkar, Benjamin Schubert, Lisa Sikkema, Avi Srivastava, Jovan Tanevski, Isaac Virshup, Philipp Weiler, Herbert B. Schiller, and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, March 2023.

[24] Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A. Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *Science*, 337(6096):816–821, August 2012.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[27] Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, and Radislav Vaisman. *Data Science and Machine Learning: Mathematical and Statistical Methods.* Chapman and Hall/CRC, November 2019.

[28] D.P. Kroese, Z.I. Botev, T. Taimre, and R. Vaisman. *Data Science and Machine Learning: Mathematical and Statistical Methods.* Chapman &

Hall/CRC machine learning & pattern recognition. CRC Press, Boca Raton, 2019.

[29] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[30] Eric S. Lander, Lauren M. Linton, Bruce Birren, Chad Nusbaum, Michael C. Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, Roel Funke, Diane Gage, Katrina Harris, Andrew Heaford, John Howland, Lisa Kann, Jessica Lehoczky, Rosie LeVine, Paul McEwan, Kevin McKernan, James Meldrim, Jill P. Mesirov, Cher Miranda, William Morris, Jerome Naylor, Christina Raymond, Mark Rosetti, Ralph Santos, Andrew Sheridan, Carrie Sougnez, Nicole Stange-Thomann, Nikola Stojanovic, Aravind Subramanian, Dudley Wyman, Jane Rogers, John Sulston, Rachael Ainscough, Stephan Beck, David Bentley, John Burton, Christopher Clee, Nigel Carter, Alan Coulson, Rebecca Deadman, Panos Deloukas, Andrew Dunham, Ian Dunham, Richard Durbin, Lisa French, Darren Grafham, Simon Gregory, Tim Hubbard, Sean Humphray, Adrienne Hunt, Matthew Jones, Christine Lloyd, Amanda McMurray, Lucy Matthews, Simon Mercer, Sarah Milne, James C. Mullikin, Andrew Mungall, Robert Plumb, Mark Ross, Ratna Shownkeen, Sarah Sims, Robert H. Waterston, Richard K. Wilson, LaDeana W. Hillier, John D. McPherson, Marco A. Marra, Elaine R. Mardis, Lucinda A. Fulton, Asif T. Chinwalla, Kymberlie H. Pepin, Warren R. Gish, Stephanie L. Chissoe, Michael C. Wendl, Kim D. Delehaunty, Tracie L. Miner, Andrew Delehaunty, Jason B. Kramer, Lisa L. Cook, Robert S. Fulton, Douglas L. Johnson, Patrick J. Minx, Sandra W. Clifton, Trevor Hawkins, Elbert Branscomb, Paul Predki, Paul Richardson, Sarah Wenning, Tom Slezak, Norman Doggett, Jan-Fang Cheng, Anne Olsen, Susan Lucas, Christopher Elkin, Edward Uberbacher, Marvin Frazier, Richard A. Gibbs, Donna M. Muzny, Steven E. Scherer, John B. Bouck, Erica J. Sodergren, Kim C. Worley, Catherine M. Rives, James H. Gorrell, Michael L. Metzker, Susan L. Naylor, Raju S. Kucherlapati, David L. Nelson, George M. Weinstock, Yoshiyuki Sakaki, Asao Fujiyama, Masahira Hattori, Tetsushi Yada, Atsushi Toyoda, Takehiko Itoh, Chiharu Kawagoe, Hidemi Watanabe, Yasushi Totoki, Todd Taylor, Jean Weissenbach, Roland Heilig, William Saurin, Francois Artiguenave, Philippe Brottier, Thomas Bruls, Eric Pelletier, Catherine Robert, Patrick Wincker, André Rosenthal, Matthias Platzer, Gerald Nyakatura, Stefan Taudien, Andreas Rump, Douglas R. Smith, Lynn Doucette-Stamm, Marc Rubenfield, Keith Weinstock, Hong Mei Lee, JoAnn Dubois, Huanming Yang, Jun Yu, Jian Wang, Guyang Huang, Jun Gu, Leroy Hood, Lee Rowen, Anup Madan, Shizen Qin, Ronald W. Davis, Nancy A. Federspiel, A. Pia Abola, Michael J.

Proctor, Bruce A. Roe, Feng Chen, Huaqin Pan, Juliane Ramser, Hans Lehrach, Richard Reinhardt, W. Richard McCombie, Melissa de la Bastide, Neilay Dedhia, Helmut Blöcker, Klaus Hornischer, Gabriele Nordsiek, Richa Agarwala, L. Aravind, Jeffrey A. Bailey, Alex Bateman, Serafim Batzoglou, Ewan Birney, Peer Bork, Daniel G. Brown, Christopher B. Burge, Lorenzo Cerutti, Hsiu-Chuan Chen, Deanna Church, Michele Clamp, Richard R. Copley, Tobias Doerks, Sean R. Eddy, Evan E. Eichler, Terrence S. Furey, James Galagan, James G. R. Gilbert, Cyrus Harmon, Yoshihide Hayashizaki, David Haussler, Henning Hermjakob, Karsten Hokamp, Wonhee Jang, L. Steven Johnson, Thomas A. Jones, Simon Kasif, Arek Kaspryzk, Scot Kennedy, W. James Kent, Paul Kitts, Eugene V. Koonin, Ian Korf, David Kulp, Doron Lancet, Todd M. Lowe, Aoife McLysaght, Tarjei Mikkelsen, John V. Moran, Nicola Mulder, Victor J. Pollara, Chris P. Ponting, Greg Schuler, Jörg Schultz, Guy Slater, Arian F. A. Smit, Elia Stupka, Joseph Szustakowki, Danielle Thierry-Mieg, Jean Thierry-Mieg, Lukas Wagner, John Wallis, Raymond Wheeler, Alan Williams, Yuri I. Wolf, Kenneth H. Wolfe, Shiaw-Pyng Yang, Ru-Fang Yeh, Francis Collins, Mark S. Guyer, Jane Peterson, Adam Felsenfeld, Kris A. Wetterstrand, Richard M. Myers, Jeremy Schmutz, Mark Dickson, Jane Grimwood, David R. Cox, Maynard V. Olson, Rajinder Kaul, Christopher Raymond, Nobuyoshi Shimizu, Kazuhiko Kawasaki, Shinsei Minoshima, Glen A. Evans, Maria Athanasiou, Roger Schultz, Aristides Patrinos, and Michael J. Morgan. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, February 2001.

[31] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, page e11517, 2023.

[32] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, July 2019.

[33] B. Müller-Hill. *The Lac Operon: A Short History of a Genetic Paradigm.* Walter de Gruyter, 1996.

[34] Emily Mullin. The era of fast, cheap genome sequencing is here, 2022.

[35] Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, Albert Xu, Marco Jost, Luke A. Gilbert, and Jonathan S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, August 2019.

[36] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, November 1901.

[37] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5–6):355–607, 2019.

[38] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[39] F Ann Ran, Patrick D Hsu, Jason Wright, Vineeta Agarwala, David A Scott, and Feng Zhang. Genome engineering using the crispr-cas9 system. *Nature Protocols*, 8(11):2281–2308, October 2013.

[40] Steven L. Salzberg. Open questions: How many genes do we have? *BMC Biology*, 16(1), August 2018.

[41] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, April 2015.

[42] Andreas Scherer. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. Wiley, October 2009.

[43] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, August 1992.

[44] B. W. Silverman and H. Läuter. Density estimation for statistics and data analysis. *Biometrical Journal*, 30(7):876–877, January 1988.

[45] Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J. Theis. Transformers in single-cell omics: a review and new perspectives. *Nature Methods*, 21(8):1430–1443, August 2024.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[47] Fei Wang, Tanveer Syeda-Mahmood, Baba C Vemuri, David Beymer, and Anand Rangarajan. Closed-form Jensen-Renyi divergence for mixture of gaussians and applications to group-wise shape registration. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, Lecture

notes in computer science, pages 648–655. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[48] Lingfei Wang, Nikolaos Trasanidis, Ting Wu, Guanlan Dong, Michael Hu, Daniel E. Bauer, and Luca Pinello. Dictys: dynamic gene regulatory network dissects developmental continuum with single-cell multi-omics. September 2022.

[49] Shuo Wang, Si-Tong Sun, Xin-Yue Zhang, Hao-Ran Ding, Yu Yuan, Jun-Jie He, Man-Shu Wang, Bin Yang, and Yu-Bo Li. The evolution of single-cell rna sequencing technology and application: Progress and perspectives. *International Journal of Molecular Sciences*, 24(3):2943, February 2023.

[50] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, January 2009.

[51] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80, December 1945.

[52] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), February 2018.

[53] Yan Wu, Esther Wershof, Sebastian M Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Kun Zhang, and Thore Graepel. Perturbench: Benchmarking machine learning models for cellular perturbation analysis, 2024.

[54] Yuanyuan Xu and Zhanjun Li. Crispr-cas systems: Overview, innovations and applications in human disease research and gene therapy. *Computational and Structural Biotechnology Journal*, 18:2401–2415, 2020.