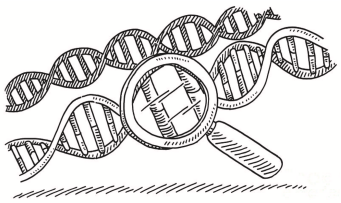
CAUSAL INFERENCE ON MULTIMODAL SINGLE-CELL DATA

LUCHO BRINKMAN

PRIMARY SUPERVISOR
THIJS VAN OMMEN

SECONDARY SUPERVISOR
GEORG KREMPL



Tackling high-dimensionality and cyclical causality to map the flow of genetic information in a cell

August 2023

Utrecht University, Faculty of Science, Artificial Intelligence. Master thesis L.H. Brinkman (5023440): *Causal Inference on Multimodal Single-Cell Data: Tackling high-dimensionality and cyclical causality to map the flow of genetic information in a cell*, supervised by Dr. M. van Ommen (primary) and Dr. ing. G.M. Krempl (secondary), August 20, 2023.

ABSTRACT

The flow of genetic information within a cell is a complex interaction between DNA, RNA, and proteins. Understanding this interplay amounts to understanding how genetic code manifests itself into cellular function, it is therefore a critical part of understanding the mechanisms of life itself. Recent technological advances have made it possible to simultaneously measure the expression of more than one of these three modalities in a single cell, thereby paving the way for machine learning to attempt the uncovering of these interactions. Algorithms for causal discovery build a graphical model to represent the causal relationships between the variables in a system. However, these methods are usually ill-suited to handle the large amount of variables and the cyclical causal relationships that are inherent to the problem at hand. Here, a two-tiered approach is proposed that first assigns the variables to partially overlapping clusters and subsequently uses an adapted interpretation of the fast causal inference (FCI) algorithm, that can handle cyclical causality, on each of the individual clusters. We investigate whether this approach can construct a collection of causal graphs that reasonably models the observed data by checking its consistency among overlapping clusters. This consistency holds up for causal graphs based on the same batch of experimental data, but not across data from distinct experiments; leading to the conclusion that this two-tiered approach opens up promising avenues to inferring cyclical causality in high-dimensional processes, but, in its current form, is yet unable to expose the intricacies of intracellular dynamics. Overall, this work can be viewed as a stepping stone towards the ultimate goal of combining all the cluster-specific graphs and creating a single high-dimensional cyclical causal graph that describes the flow of genetic information in a cell.

CONTENTS

ACRONYMS

DNA	deoxyribonucleic acid
RNA	ribonucleic acid
mRNA	messenger-RNA
cDNA	complementary-DNA
scRNA-Seq	single-cell RNA-sequencing
ATAC-Seq	assay for transposase-accessible chromatin using sequencing
scATAC-Seq	single-cell ATAC-Seq
ADT	antibody derived tag
CITE	cellular indexing of transcriptomes and epitomes by sequencing
MULTI	chromium single-cell multiome ATAC-Seq + gene expression
DAG	directed acyclic graph
PDAG	partially directed acyclic graph
CPDAG	completed PDAG
DPAG	directed partial ancestral graph
PC algorithm	Peter Spirtes & Clark Glymour's algorithm
GES	greedy equivalence search
FES	forward equivalence search
BES	backward equivalence search
FGS	fast greedy search
FCI	fast causal inference
BIC	Bayesian information criterion
DSB	denoised and scaled by background
TF-IDF	term frequency inverse document frequency
SVD	singular value decomposition
PCA	principal component analysis
DMG	directed mixed graph
ADMG	acyclic DMG
DG	directed graph
DMAG	directed maximal ancestral graph

DPAG	directed partial ancestral graph
CDPAG	complete DPAG

1

INTRODUCTION

1.1 THE FLOW OF INFORMATION IN A CELL

All living things are made up from cells, these are the basic building blocks of any organism. Some bacteria consist of only a single cell, while complex mammals may be made up of vast amounts of them. The human body, for example, is estimated to count as many as 37 trillion cells (Bianconi et al. 2013). To a large extend the function and behavior of these cells is determined by the proteins that float within it. Proteins fulfill many different roles within the ecosystem of a cell, among which are the duplication of DNA during cell division and the transport of molecules to various locations. However, their primary task is to catalyze chemical reactions. By enabling certain reactions, but not others, proteins dictate the metabolism of the cell (Clancy and Brown 2008).

Proteins are encoded for in the genes of the deoxyribonucleic acid (DNA) of an organism. The expression of these genes, meaning: the production of the protein corresponding to that gene, is done via ribonucleic acid (RNA). The information contained in a DNA gene is copied to a messenger-RNA (mRNA) molecule during a process called *transcription*. This piece of mRNA is then used to create the intended protein during the process of *translation*.

In other words, the flow of genetic information within a cell is governed by roughly three modalities that interact with each other. At the highest level is DNA, this can be transcribed to the second level: RNA, which can then be translated into the third level: proteins. The study of the interaction between these layers is crucial in understanding how cells, and by extension how life, operates.

1.2 MULTIMODAL SINGLE-CELL DATA

Measurement of these three modalities present a number of difficulties. Traditional methods for sequencing DNA or RNA measure the occurrences in mixtures of millions of cells (e.g. Shendure and Ji 2008). A substantial drawback of these methods is that information on individual cells is lost, the investigation of cellular function and cell diversity is nigh-on impossible with aggregated data only. Enter: single-cell sequencing, where measurements can be taken from a single cell. Since the first single-cell transcriptome by Tang et al. 2009, multiple techniques for these deep-level measurement have been developed and innovated upon. Nowadays it is possible to perform single-cell measurements on large quantities of cells simultaneously (Klein et al. 2015, Macosko et al. 2015).

However, in order to study the interplay between modalities it is not enough to have a single modality measured in a single cell, to enable this an additional innovation was needed. This

came with Stoeckius et al. 2017 and Cao et al. 2018, who introduced the first procedures to measure multiple modalities in single cells. Commercial kits for both methods came out two years later in 2019 and 2020 respectively, opening the door for data scientists to unravel the relationships between the many DNA genes, RNA molecules, and proteins floating around in a cell. This is one of the open problems in the industry, and it is the focal point of this thesis.

1.3 CAUSAL INFERENCE

The field of machine learning provides ample angles to approach this problem from. For example, the approaches in Lance et al. 2021 focus around deep learning to predict and match modalities and to summarize cellular identity from observed data. Our investigation, however, is centered around causal inference models.

Studying causality is important for anyone who is interested in questions like *how* did these various causes influence their effect, and *why* did the consequence manifest itself the way it did. To satisfyingly answer these questions it is not enough to study correlation; The observed negative correlation between ice-cream sales and household gas consumption in the Netherlands does not imply that buying ice-cream causes people to consume less gas, nor vice versa. More likely, the two phenomena share a common cause. In this sense, the study of causality is an enrichment to the age-old field of classical statistics that tends to deal with correlation.

A principled approach to investigating causality in a system is the construction of a directed mixed graph (DMG) that represents the conditional dependencies between the various variables, represented as nodes in the graph. In the current investigation these variables are the DNA genes, RNA molecules, and proteins present in a cell. If accurate, such a causal graph, in combination with a set of functions, can be used to make the same predictions as those that Lance et al. focus on. However, it tries to answer the much more general open question of how all the different genetic entities interact with each other to produce the specific metabolism that a cell exhibits.

Sachs et al. 2005 successfully attempted a similar investigation using single-cell protein data. Note that their research predates the existence of single-cell DNA and RNA data, let alone multimodal single-cell data. The data that is available now gives a much more fine-grained view of the cellular processes, but it also introduces new challenges. While Sachs et al. constructed a causal graph where the nodes represented the few dozen measured proteins, the current data would give rise to a causal graph with hundreds of thousands of nodes, this is a degree of dimensionality that standard causal inference algorithms cannot handle. Some recent work like Ramsey et al. 2017 have shown the efficacy of algorithms that search through the space of equivalence classes of directed acyclic graphs (DAGs), instead of searching through the space of DAGs directly. These, and similar, methods could potentially handle the construction of the high-dimensional causal graphs that are needed for multimodal single-cell data.

The second major challenge of multimodal single-cell data for causal inference is the cyclical nature of many of the causal relationships in this network. In the simplest case of a negative feedback loop, one could think of a gene that promotes the production of a certain protein. Once the protein in ubiquitous in the cell, the accessibility of that gene should be inhibited to prevent further production, creating a direct cycle of causality. Causal inference algorithms do not generally allow such cycles, certainly not the algorithms that are equipped to handle the

high-dimensional systems. However, for low-dimensional graphs solutions exist to represent cyclical relationships (Forré and Mooij 2018, Mooij and Claassen 2020).

1.4 RESEARCH GOALS

In order to tackle the problems outlined above, we propose a two-tiered approach to find the causal relationships between the numerous interaction in a cell. First we use a custom clustering algorithm to assign all variables to one or more clusters. The goal is to *(i)* create clusters within which the variables are heavily dependent on each other; *(ii)* make sure that the clusters are relatively small in size; and *(iii)* retain some partial overlap between clusters.

In the second tier we take each cluster and consider it independently of the rest of the nodes. Using the resulting low-dimensional dataset a causal graph is constructed with a causal inference algorithm that allows for cyclical causal relations.

The central question of this thesis is, whether this process can be used to create a causal graph that accurately models the observed data. If that is the case, it could be a general approach for high-dimensional cyclical causal inference. Our data, however, is hard to interpret, noisy, and high-dimensional, consequently the validation of results is a difficult task.

The main source of validation will be achieved by checking whether the overlapping parts between clusters imply consistent causal relationships. Admittedly, this is by no means a rigorous assessment, but it should provide a rough indication of the merit of this method. Additionally this consistency would indicate whether or not our procedure could be used as a first step towards creating a single high-dimensional cyclical causal graph by recombining all the cluster-specific causal graphs into one.

2

LITERATURE REVIEW

Here we discuss the existing literature that provides the theoretical basis of our work. First we cover some background knowledge about the data that we are working with; then we go over the needed definitions and theory concerning causal graphs; and last, we discuss some of the key causal inference algorithms and their traits.

2.1 MULTIMODAL SINGLE-CELL DATA

The collection of multimodal single-cell data is a relatively new, and non-trivial, endeavor; it is therefore useful to have a certain level of understanding of how the data was produced in the laboratory, and be aware of any resulting limitations. Here we look at two different procedures for measuring this type of data. The first: CITE, it measures two modalities, namely RNA and proteins. The second: MULTI, also measures two modalities, namely RNA and DNA.

2.1.1 *CITE: measuring RNA and proteins from a single cell*

The cellular indexing of transcriptomes and epitomes by sequencing (CITE) procedure was first proposed by Stoeckius et al. 2017. It is a single-cell RNA-sequencing (scRNA-Seq) method that simultaneously extracts information of the proteins on the surface of the cell for which antibodies are available.

The procedure begins by preparing a mixture of antibody derived tags (ADTs), these are molecules that bind to specific proteins, and are labeled with a barcode to later identify which antibody it is, by extension this can be used to identify to which protein it is bound. The cells that are to be analyzed are first subjected to this mixture, during which the ADTs bind to the proteins on the surface of the cells.

Using the Drop-Seq procedure, the cells can be individually extracted from this mixture along with a bead. This bead is important because each bead is covered by uniquely barcoded DNA, these barcodes can later be used to identify which RNA molecules and proteins all came from the same cell. Next, the membrane of the cell is broken down causing the RNA from the cell as well as the ADTs on the surface to move freely and to form complementary-DNA (cDNA) with the barcoded molecules from the bead.

Once all cells have undergone this procedure, the sequencing can begin. Since all RNA molecules have a barcode that is readable by sequencing and that maps to a single cell, and since all ADTs also have this in addition to a barcode identifying to which protein it was bound, the sequence data can be labelled correctly.

The limitations of this method are that no measurement is taken for the proteins within a cell, nor for the proteins for which no ADT is prepared or even exists. Additionally, CITE inherits the limitations of the more general scRNA-Seq method, this includes: *(i)* a high degree of noise, and *(ii)* difficulty measuring genes that are expressed only in low quantities.

2.1.2 *MULTI: measuring DNA and RNA from a single cell*

A large part of the DNA in a cell is usually tightly wrapped up, and is therefore unable to be transcribed into RNA. Consequently, the parts of the DNA that are not wrapped up, the open parts, are important for the function of a cell. The assay for transposase-accessible chromatin using sequencing (ATAC-Seq) technique cuts up these open parts to expose them to sequencing.

The chromium single-cell multiome ATAC-Seq + gene expression (MULTI) method, proposed by Cao et al. 2018, combines scRNA-Seq, with a single-cell ATAC-Seq (scATAC-Seq) procedure, which measures the accessible parts of the DNA using ATAC-Seq while simultaneously barcoding them uniquely per cell.

Combining the two sequencing methods starts with isolating the cell cores, the nuclei, from the rest of the cells. This can be done in bulk, resulting in a mixture that contains all the nuclei; these can then be extracted from this mixture one by one using the same Drop-Seq procedure used by CITE. During this process, each nucleus is accompanied by a uniquely barcoded bead.

Next, the isolated nucleus is broken down, exposing the accessible pieces of DNA so that they can be cut up and bound to the barcoded molecules from the bead; simultaneously, these same barcoded molecules also bind to the RNA floating around in the nucleus. Once all the nuclei have undergone this procedure all pieces of RNA and DNA are labeled such that, during sequencing, it can be identified to which cell they belonged.

Similar to the CITE procedure, the MULTI procedure also inherits the limitations of scRNA-Seq. Apart from that it should be noted that only RNA data from the cell nucleus is being measured, RNA that floated in the rest of the cell is invisible to this method.

2.2 CAUSAL GRAPHS: NOTATION AND DEFINITIONS

Before going over the various causal inference algorithms, we should discuss the different types of causal graphs that they use. Generally, a graph is a combination of nodes and edges between those nodes, but different types of graphs can allow different types of edges, and may be interpreted in different ways. Throughout this section we primarily keep to the notations and conventions maintained by Mooij and Claassen 2020. We start by defining the different types of edges, moving on to the different types of graphs, and finishing with the types of graphs that themselves represent collections of graphs.

2.2.1 *Edges*

For a graph $\mathcal{G}$, its collection of nodes is denoted by $\mathcal{V}$. Throughout this work self-cycles are not allowed, so an edge is a connection between two distinct nodes $v, w \in \mathcal{V}$ and can belong to one of the following type-collections.

- *Undirected edges (—)*: Denoted as: $\mathcal{U} = \{\{v, w\} \mid v, w \in \mathcal{V},\ v \neq w\}$.

- *Directed edges ($\rightarrow$)*: Denoted as: $\mathcal{E} = \{(v, w) \mid v, w \in \mathcal{V},\ v \neq w\}$.
- *Bidirected edges ($\leftrightarrow$)*: Denoted as: $\mathcal{F} = \{\{v, w\} \mid v, w \in \mathcal{V},\ v \neq w\}$.
- *Directed open-circle edges ($\circ\!\!\rightarrow$)*: Denoted as: $\mathcal{D} = \{(v, w) \mid v, w \in \mathcal{V},\ v \neq w\}$.
- *Undirected open-circle edges ($\circ\!\!-\!\!\circ$)*: Denoted as: $\mathcal{C} = \{\{v, w\} \mid v, w \in \mathcal{V},\ v \neq w\}$.

Additionally we may use an asterisk ($*$) instead of an edge-end to denote any type of edge-end that a specific graph allows. Edges with such ends are never part of any actual graph, but may be used colloquially to describe them. For example, the edge $*\!\!\rightarrow$ in a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ can be either $\rightarrow$, $\leftrightarrow$, or $\circ\!\!\rightarrow$.

2.2.2 *Graphs*

Figure 1a represents the relationships between the different graphs that we consider here. Starting with the most general, a directed mixed graph (DMG) is a graph that allows directed edges and bidirected edges, that is: $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$. The interpretation of a directed edge $v \rightarrow w$ is that v causes w, whereas a bidirected edge $v \leftrightarrow w$ means that there exists a latent confounder that causes both v and w. Multiple edges between nodes are allowed as long as they do not create duplicates in either $\mathcal{E}$ or $\mathcal{F}$.

The following definitions for a DMG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$ will be useful.

- *Set of parents*: $\text{PA}_{\mathcal{G}}(v) := \{w \in \mathcal{V} \mid w \rightarrow v \in \mathcal{E}\}$.
- *Adjacency*: Two nodes $v, w \in \mathcal{V}$ are considered to be adjacent if $v \rightarrow w \in \mathcal{E}$, $v \leftarrow w \in \mathcal{E}$, or $v \leftrightarrow w \in \mathcal{F}$. Denote the *set of adjacencies*: $\text{AD}_{\mathcal{G}}(v) := \{w \in \mathcal{V} \mid v \text{ and } w \text{ are adjacent in } \mathcal{G}\}$.
- *Walk*: A tuple $\langle v_0, e_1, ..., e_n, v_n \rangle$ of alternating nodes $v_0, ..., v_n \in \mathcal{V}$ and edges $e_1, ..., e_n \in \mathcal{E} \cup \mathcal{F}$, where e_k is an edge between v_{k-1} and v_k, is called a walk between v_0 and v_n.
- *Path*: A walk $\langle v_0, e_1, ..., e_n, v_n \rangle$ where all nodes $v_0, ..., v_n$ are distinct, is called a path between v_0 and v_n.
- *Directed walk/path*: A walk or path $\langle v_0, e_1, ..., e_n, v_n \rangle$ is called directed if $e_k = v_{k-1} \rightarrow v_k$ for $k = 1, ..., n$.
- *Trivial walk/path*: A walk or path is called trivial if it contains only 1 node and no edges.
- *Collider*: Let $\pi = \langle v_0, e_1, ..., e_n, v_n \rangle$ be a walk on $\mathcal{G}$, then v_k is a collider on π if e_k is of the form $v_{k-1} *\!\!\rightarrow v_k$ and e_{k+1} is of the form $v_k \leftarrow\!\!* v_{k+1}$ with $k \in \{1, ..., n-1\}$.
  - By extension, a *non-collider* is any node on π that is not a collider.
- *Directed cycle*: A directed path from v to w combined with $w \rightarrow v \in \mathcal{E}$ is a directed cycle.
- *Almost directed cycle*: A directed path from v to w combined with $w \leftrightarrow v \in \mathcal{E}$ is an almost directed cycle.
- *Acyclic*: $\mathcal{G}$ is called acyclic if it contains no directed cycles.
- *Set of ancestors*: $\text{AN}_{\mathcal{G}}(v) := \{w \in \mathcal{V} \mid \mathcal{G} \text{ has a (possibly trivial) directed path from w to v}\}$.
  - By extension for $V \subseteq \mathcal{V}$: $\text{AN}_{\mathcal{G}}(V) := \cup_{v \in V} \text{AN}_{\mathcal{G}}(v)$.
- *Set of descendants*: $\text{DE}_{\mathcal{G}}(v) := \{w \in \mathcal{V} \mid \mathcal{G} \text{ has a (possibly trivial) directed path from v to w}\}$.
  - By extension for $V \subseteq \mathcal{V}$: $\text{DE}_{\mathcal{G}}(V) := \cup_{v \in V} \text{DE}_{\mathcal{G}}(v)$.

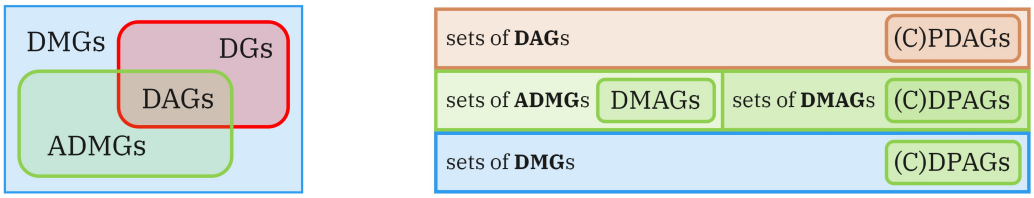
- *Strongly connected component*: $\mathrm{sc}_{\mathcal{G}}(v) := \mathrm{an}_{\mathcal{G}}(v) \cap \mathrm{de}_{\mathcal{G}}(v)$.
  - By extension for $V \subseteq \mathcal{V}$: $\mathrm{sc}_{\mathcal{G}}(V) := \cup_{v \in V} \mathrm{sc}_{\mathcal{G}}(v)$.
- d-*blocked*: A walk $\pi = \langle v_0, e_1, ..., e_n, v_n \rangle$ is d-blocked in $\mathcal{G}$ by $C \subseteq \mathcal{V}$ if at least one of the following three conditions is met:
  1. $v_0 \in C$ or $v_n \in C$.
  2. π contains a collider v_k, for which $v_k \notin \mathrm{an}_{\mathcal{G}}(C)$.
  3. π contains a non-collider v_k for which: $v_k \in C$.
- σ-*blocked*: A walk $\pi = \langle v_0, e_1, ..., e_n, v_n \rangle$ is σ-blocked in $\mathcal{G}$ by $C \subseteq \mathcal{V}$ if at least one of the following three conditions is met:
  1. $v_0 \in C$ or $v_n \in C$.
  2. π contains a collider v_k, for which $v_k \notin \mathrm{an}_{\mathcal{G}}(C)$.
  3. π contains a non-collider v_k for which: $v_k \in C$ and at least one of the following two conditions is met:
     a) $e_{k+1} = v_k \rightarrow v_{k+1}$ and $v_{k+1} \notin \mathrm{sc}_{\mathcal{G}}(v_k)$
     b) $e_k = v_{k-1} \leftarrow v_k$ and $v_{k-1} \notin \mathrm{sc}_{\mathcal{G}}(v_k)$
- d/σ-*separation*: For $s \in \{d, \sigma\}$: subsets $A, B \subseteq \mathcal{V}$ are s-separated by $C \subseteq \mathcal{V}$ if every path in $\mathcal{G}$ between any node $a \in A$ and any node $b \in B$ is s-blocked by C. Notation: $A \perp^s_{\mathcal{G}} B \mid C$.
  - If A and B are not s-separated by C, then they are said to be s-*connected* given C.
- d/σ-*independence model*: For $s \in \{d, \sigma\}$: the set of all s-separations is called the s-independence model of $\mathcal{G}$: $\mathbb{IM}_s(\mathcal{G}) := \{\langle A, B, C \rangle \mid A, B, C \subseteq \mathcal{V},\ (A \perp^s_{\mathcal{G}} B \mid C)\}$.
- d/σ-*Markov equivalence*: For $s \in \{d, \sigma\}$: two DMGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are considered to be s-Markov equivalent if $\mathbb{IM}_s(\mathcal{G}_1) = \mathbb{IM}_s(\mathcal{G}_2)$.

A DMG that is acyclic is also called an acyclic DMG (ADMG). Note that for these graphs, σ-separation reduces to d-separation, since strongly connected components loosely describe nodes that are tied together in directed cycles. More precisely, for acyclic graphs, the following holds: $\mathrm{sc}_{\mathcal{G}}(v)=\{v\}$ for all $v \in V$. The notion of σ-separation and its derived concepts are extensions of the d-separation concepts, they are designed to account for the presence of cyclical causal relationships.

If a DMGs has no bidirected edges, it is called a directed graph (DG). The absence of bidirected edges means that these graphs are assumed to contain no latent confounders, that is, all relevant variables in the causal network are assumed to be observed in the data. A graph that is both an ADMG and a DG is called a directed acyclic graph (DAG). Note that by excluding bidirected edges and by excluding cycles (among which cycles of size 2: $v \rightarrow w \rightarrow v$), a DAG has at most one edge between any two nodes.

2.2.3 *Graphs of sets of graphs*

Typically, causal inference algorithms are not able to return one specific causal graph, but rather a set of possible causal graphs that is implied by the data. Think, for example, of two graphs that are d-Markov equivalent, given just the data and no further contextual knowledge, it would be impossible to favor one over the other. However, these sets of graphs are in turn



(a) Venn-diagram depicting types of causal graphs.

(b) Set-element relationships for the various graphs of sets of graphs.

Figure 1: Relationships between the different types of graphs and graphs of sets of graphs.

often represented as graphs themselves. In this section we will cover these kinds of graphs to prepare ourselves for the ensuing algorithms that produce them.

A simple example of a graph that represents a set of graphs is the partially directed acyclic graph (PDAG). This is a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{U} \rangle$ with both directed ($\leftarrow$, $\rightarrow$) and undirected (—) edges. By interpreting any undirected edge v—w as possibly being either of the form $v \rightarrow w$ or $v \leftarrow w$, it represents any DAG that can be constructed by orienting every undirected edge one way or the other without violating the condition of acyclicity. In other words, a PDAG represents a set of DAGs.

To describe the significance of a completed PDAG (CPDAG) we first introduce the following two definitions.

- *Skeleton*: The skeleton of any type of graph is an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{U} \rangle$ with the same nodes as the original graph and an undirected (—) edge between any two adjacent nodes in the original graph.
- *Equivalence class*: The equivalence class of a DAG $\mathcal{G}$ is the set of all DAGs that are d-Markov equivalent to $\mathcal{G}$.
- *v-structure*: A triplet $\langle v_1, v_2, v_3 \rangle$ is called a v-structure in PDAG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{U} \rangle$ if and only if $v_1, v_2, v_3 \in \mathcal{V}$ and $v_1 \rightarrow v_2 \in \mathcal{E}$ and $v_2 \leftarrow v_3 \in \mathcal{E}$ and v_1 and v_3 are not adjacent in $\mathcal{G}$.

Verma and Pearl 1990 have shown that two DAGs are equivalent if and only if they have the same skeleton and the same v-structures. An equivalence class can thus be represented by its CPDAG, which is a PDAG that consists of a skeleton where the edges of the v-structures have been oriented as well as any edges whose orientation is implied by the conditions that no cycles and no extra v-structure may be introduced. As such, a CPDAG uniquely identifies an entire equivalence class of DAGs, Figure 1b gives an overview of the relationships between these different graph-types.

Now we move on to the introduction of some useful types of graphs that describe sets of ADMGs. In order to do so, we first define some notions that will come in handy in the description of these graphs.

- d-*inducing path*: A path $\pi = \langle v_0, e_1, ..., e_n, v_n \rangle$ on ADMG $\mathcal{G}$, is called a d-inducing path between v_0 and v_n if $v_k \in \text{AN}_{\mathcal{G}}(\{v_0, v_n\})$ and v_k is a collider on π for k=1, ..., n–1.
- σ-*inducing path*: A path $\pi = \langle v_0, e_1, ..., e_n, v_n \rangle$ on DMG $\mathcal{G}$, is called a σ-inducing path between v_0 and v_n if $v_k \in \text{AN}_{\mathcal{G}}(\{v_0, v_n\})$ and any directed edge on π pointing out of v_k points to an element of $\text{sc}_{\mathcal{G}}(v_k)$ for k=1, ..., n–1.
- *Maximal*: If an ADMG contains no d-inducing paths between any two nodes that are not adjacent to each other, then the graph is said to be maximal.

- *Ancestral*: A DMG $\mathcal{G}$ is said to be ancestral if it contains no directed or almost directed cycles.

Using the above definitions, an ADMG is called a directed maximal ancestral graph (DMAG) if it is *(i)* maximal, *(ii)* ancestral, and *(iii)* contains no more than one edge between any pair of distinct nodes.

Since a d-inducing path between two nodes implies that they cannot be d-separated by any conditioning set (not containing the nodes themselves), the *maximal* condition on a DMAG $\tilde{\mathcal{G}}$ means that any two nodes that are always d-connected have to be adjacent in $\tilde{\mathcal{G}}$.

A procedure proposed by Richardson and Spirtes 2002 creates a DMAG from an arbitrary ADMG $\mathcal{G}$, this is called the DMAG *induced* by $\mathcal{G}$: $\text{DMAG}(\mathcal{G})$ and is constructed as follows.

1. For $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$, let $\text{DMAG}(\mathcal{G}) = \langle \tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{\mathcal{F}} \rangle$ with $\tilde{\mathcal{V}} = \mathcal{V}$.
2. Create an edge between two distinct nodes $v, w \in \tilde{\mathcal{V}}$ if and only if the original graph $\mathcal{G}$ has an inducing path between v and w. The type and orientation of the edge should be: $v \to w$ if $v \in \text{AN}_{\mathcal{G}}(w)$, $v \leftarrow w$ if $w \in \text{AN}_{\mathcal{G}}(v)$, and $v \leftrightarrow w$ otherwise.

The resulting DMAG maintains the same information with regards to the absence and presence of ancestral relations and d-separation, and we can now interpret a DMAG $\tilde{\mathcal{G}}$ as the set of ADMGs $\mathcal{G}$ for which $\text{DMAG}(\mathcal{G}) = \tilde{\mathcal{G}}$.

Taking yet a further step towards generalization, we can define a directed partial ancestral graph (DPAG) as a graph $\mathcal{P} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ that, in addition to directed and bidirected edges, can also contain directed open-circle edges and undirected open-circle edges. That is, an edge in $\mathcal{P}$ can be of type: $\to$, $\leftarrow$, $\leftrightarrow$, $\circ\!\!\to$, $\leftarrow\!\!\circ$, or $\circ\!\!-\!\!\circ$. Apart from that, the same conditions as for a DMAG should hold: $\mathcal{P}$ is *(i)* maximal, *(ii)* ancestral, and *(iii)* contains at most one edge between two distinct nodes.

The interpretation of any open-circle edge-end is that it can either be an arrowhead or an arrowtail. As such, it identifies the set of all DMAGs that can be created by resolving every open-circle edge-end in the DPAG to either an arrowhead or tail, without violating the DMAG-constraints. Because every DMAG, in turn, represents a set of ADMGs, we can also say that an ADMG $\mathcal{G}$ is *contained* in $\mathcal{P}$ if $\mathcal{P}$ contains $\text{DMAG}(\mathcal{G})$.

Since a DMAG $\tilde{\mathcal{G}}$ retains all the d-separations and connections of the ADMGs that it is induced by, we can say that its independence model $\mathbb{M}_{\text{d}}(\tilde{\mathcal{G}})$ equals the independence model $\mathbb{M}_{\text{d}}(\mathcal{G})$ of any $\mathcal{G}$ with $\text{DMAG}(\mathcal{G}) = \tilde{\mathcal{G}}$. Using this definition we say that, if a DPAG $\mathcal{P}$ contains $\tilde{\mathcal{G}}$ and an edge-end is identified (i.e. not an open-circle) in $\mathcal{P}$ if and only if it is identifiable from $\mathbb{M}_{\text{d}}(\tilde{\mathcal{G}})$, then $\mathcal{P}$ is called the complete DPAG (CDPAG) of $\tilde{\mathcal{G}}$. By extension $\mathcal{P}$ is then also the CDPAG of any ADMG $\mathcal{G}$ that induces $\tilde{\mathcal{G}}$. As such, a CDPAG represents a d-Markov equivalence class of ADMGs.

Figure 1b provides an overview of the graphs that represent sets of graphs that we have discussed here. An important takeaway from this all is that CPDAGs and CDPAGs are specific cases of PDAGs and DPAGs respectively that represent d-Markov equivalence classes, while in general PDAGs and DPAGs do not.

2.2.4 *Acyclification*

A (possibly cyclical) DMG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$ can be transformed into an ADMG $\mathcal{G}' = \langle \mathcal{V}', \mathcal{E}', \mathcal{F}' \rangle$ by way of the following *acyclification* procedure.

1. Let $\mathcal{G}'$ have the same nodes as $\mathcal{G}$, $\mathcal{V}' := \mathcal{V}$.
2. For all edges $v \to w \in \mathcal{E}$ with $v \notin \mathrm{sc}_{\mathcal{G}}(w)$, let $v \to w' \in \mathcal{E}'$ for all $w' \in \mathrm{sc}_{\mathcal{G}}(w)$.
3. For all edges $v \leftrightarrow w \in \mathcal{F}$ with $v \notin \mathrm{sc}_{\mathcal{G}}(w)$, let $v \leftrightarrow w' \in \mathcal{F}'$ for all $w' \in \mathrm{sc}_{\mathcal{G}}(w)$.
4. For all pairs of distinct nodes $v, w \in \mathcal{V}$ with $v \in \mathrm{sc}_{\mathcal{G}}(w)$, let one of the following hold: $v \to w \in \mathcal{E}'$, $v \leftarrow w \in \mathcal{E}'$, or $v \leftrightarrow w \in \mathcal{F}'$.

This process can be described as copying all edges (directed and bidirected) pointing into a strongly connected component to also point to all other nodes in that strongly connected component, and additionally, making any strongly connected component fully connected. This last part can be done in multiple ways, so acyclifications are generally not unique. Also note that this procedure potentially creates a lot of extra edges which should not be interpreted as representing causal relationships, instead, the point of these edges is to maintain the same σ-separations in $\mathcal{G}'$ as in $\mathcal{G}$.

This leads us to the reason why we might be interested in an arbitrary acyclification $\mathcal{G}'$ of DMG $\mathcal{G}$, which is that $\mathbb{M}_\sigma(\mathcal{G}) = \mathbb{M}_d(\mathcal{G}')$ and there exists a σ-inducing path between two nodes v and w in $\mathcal{G}$ if and only if there exists a d-inducing path between the two nodes in $\mathcal{G}'$ (Mooij and Claassen 2020). Via this interpretation we can define, for DMG $\mathcal{G}$, the set of all ADMG acyclifications of $\mathcal{G}$, and denote it as: $\mathrm{ACY}(\mathcal{G})$.

From the previous section, a DPAG $\mathcal{P}$ is a graph that, through DMAGs, represents a set $\mathcal{X}$ of ADMGs. However, without changing the definition of its construction, we can also interpret it to additionally contain any DMG $\mathcal{G}$ for which all acyclifications are contained in $\mathcal{X}$. That is, $\mathcal{P}$ is interpreted as representing a set of DMGs: $\mathcal{X} \cup \{\mathcal{G} \mid \mathrm{ACY}(\mathcal{G}) \subseteq \mathcal{X}, \mathcal{G} \text{ is a DMG}\}$.

Mooij and Claassen 2020 show that for a DMG $\mathcal{G}$ contained by DPAG $\mathcal{P}$ the following holds:

- Two nodes v and w are adjacent in $\mathcal{P}$ if and only if $\mathcal{G}$ has a σ-inducing path between v and w.
- If $\mathcal{P}$ has edge: $v \leftarrow\!\ast w$ then $v \notin \mathrm{AN}_{\mathcal{G}}(w)$.
- If $\mathcal{P}$ has edge: $v \to w$, then $v \in \mathrm{AN}_{\mathcal{G}}(w)$.

Note that if $\mathcal{G}$ is an ADMG, then this interpretation is consistent with the interpretation from the previous section, that is, $\mathcal{P}$ contains the DMAG $\mathrm{DMAG}(\mathcal{G})$.

The latter two points concerning (non-)ancestry are useful, but it is possible to make stronger assertions about these properties. In order to make those, we first define the following two properties for a DPAG $\mathcal{P}$.

- *Possibly directed path*: A path $\langle v_0, e_1, ..., e_n, v_n \rangle$ between v_0 and v_n in $\mathcal{P}$ is called possibly directed if e_i is of the form $v_{i-1} \to v_i$, $v_{i-1} \circ\!\!\to v_i$, or $v_{i-1} \circ\!\!-\!\!\circ v_i$ for $i = 1, ..., n$.
- *Uncovered path*: A path $\langle v_0, e_1, ..., e_n, v_n \rangle$ between v_0 and v_n in $\mathcal{P}$ is called uncovered if v_{i-2} and v_i are not adjacent in $\mathcal{P}$ for $i = 2, ..., n$.

With these definitions in hand we can posit the stronger results by Mooij and Claassen 2020 about identifiable (non-)ancestry of a DMG contained by a DPAG $\mathcal{P} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$.

- If $\mathcal{P}$ is a CDPAG and for $v, w \in \mathcal{V}$ either of the following two holds: *(i)* there is a directed path from v to w in $\mathcal{P}$, or *(ii)* $\langle v, e_1, u, e_2, ..., e_n, w \rangle$ and $\langle v, e'_1, u', e'_2, ..., e'_n, w \rangle$ are

uncovered possibly directed paths from v to w, where $u \neq u'$ and $u \notin \text{AD}_{\mathcal{P}}(u')$; then $v \in \text{AN}_{\mathcal{G}}(w)$.

- For $v, w \in \mathcal{V}$, if there exists no possibly directed path from v to w in $\mathcal{P}$ then $v \notin \text{AN}_{\mathcal{G}}(w)$.

The generalization of CDPAG to interpret possibly cyclical DMGs finishes up our coverage of different types of graphs that represent sets of causal graphs. As a reference, Figure 1b summarizes the various relationships.

2.3 CAUSAL INFERENCE

Finding a correct DMG to represent a system of interacting variables comes with a range of problems. Indeed the number of possible DMGs for a system with n variables quickly becomes astronomically large. If we allow for latent confounders and place no restrictions on cyclicity, then for each of the $\frac{1}{2}n(n-1)$ pairs of nodes, there could be a directed edge in either direction (or both) as well as a bidirected edge between them, resulting in $8^{\frac{1}{2}n(n-1)}$ possible DMGs. Even for a system with the modest amount of 10 variables this already amounts to over 10^{40} possibilities.

Clearly, efficient algorithms are needed if we are even to attempt a search like this. First we discuss Peter Spirtes & Clark Glymour's algorithm (PC algorithm) which is the basic starting point of most causal inference problems. As such, it nicely introduces some of the core ideas of the field, and additionally it illustrates the two main challenges of our own investigation. Next we will go over some algorithms that can handle the high-dimensional data that we are dealing with, fundamentally this is the GES algorithm, with the FGS extension of it. After that we will turn to the FCI algorithm because it provides a solution to the problem of latent confounders, and additionally it provides a good segue into its extensions that deal with cyclical causality.

2.3.1 *PC algorithm*

The PC algorithm, proposed by Spirtes and Glymour 1991 assumes that the input data was generated by a DAG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, it consists of three parts. First a skeleton graph is constructed, then the v-structures are identified, and finally edges whose orientations are now implied by the DAG-assumption are set accordingly. These steps build a PDAG $\mathcal{H} = \langle \mathcal{V}, \tilde{\mathcal{E}}, \tilde{\mathcal{U}} \rangle$, with the same vertex set as $\mathcal{G}$ that should represent (among others) the DAG $\mathcal{G}$ that generated the data. The main ideas of the three steps are as follows.

1. SKELETON CONSTRUCTION: The construction of the skeleton uses that if two variables $v, w \in \mathcal{V}$ are independent of each other conditional on any $C \subseteq \text{AD}_{\mathcal{G}}(v) \setminus \{w\}$, then there is no direct causal link between the two, though an indirect causal relation may still exist.

   Concretely, we start by instantiating $\mathcal{H}$ as the fully connected undirected graph. Then, for $n = 0, 1, ..., n-2$ we iteratively do the following:

   - Check for every edge v—w in $\mathcal{H}$ whether v and w are independent of each other conditional on any subset $C \subseteq \text{AD}_{\mathcal{H}}(v) \setminus \{w\}$ of size n. If so, we *(i)* we remove v—w from $\mathcal{H}$, and *(ii)* store C in memory.

   The resulting trimmed undirected graph is the skeleton and is used as input for the next step. By trimming down the graph iteratively, we see to it that each iteration gen-

erally has less variable pairs that need to be tested. This is significant since conditional independence tests become less accurate with larger conditioning sets.

II. V-STRUCTURE IDENTIFICATION: For each connected triplet of nodes v_1—v_2—v_3 in $\mathcal{H}$ with $v_1 \notin \text{AD}_{\mathcal{H}}(v_3)$, we check whether v_2 was in the conditioning set that was recorded in the previous step for v_1 and v_3. If this is the case, we leave the edges undirected, otherwise we orient the triplet as a v-structure: $v_1 \rightarrow v_2 \leftarrow v_3$.

III. IMPLIED EDGE ORIENTATION: After the previous step, we can assume that all v-structures have been identified. Any still undirected edges that would create a new v-structure when oriented in a certain direction, can thus be oriented in the opposite direction.

Similarly the PC algorithm assumes only acyclic causal relationships, so any undirected edge that would create a directed cycle if oriented one way, should be oriented the other way.

The result is a PDAG $\mathcal{H}$ which, by virtue of the steps I and II has locked in its skeleton and v-structures; subsequently, step III completes it, making it a CPDAG.

2.3.2 *Greedy equivalence search*

Performing (conditional) independence tests is an intensive computational task, and although step I of the PC algorithm is designed to avoid unnecessary calculations, this nevertheless becomes an unfeasible endeavor for systems with large amounts of variables. The greedy equivalence search (GES) proposed by Chickering 2002 combines two main ideas to improve on this. Both are connected to the idea of calculating a decomposable score to judge a graph's fitness to represent the data. Here, *decomposable* means that it is the sum of parts, where each part is determined by only a single node and its parents.

The first idea concerns this score in relationship to equivalence classes of DAGs as discussed in section 2.2.3. Since two d-Markov equivalent DAGs fit any body of data equally well, we know that their score should also be the same. Knowing that the scores of all DAGs in the same equivalence class are the same, we can significantly increase efficiency by searching the space of equivalence classes of DAGs instead of searching the space of individual DAGs which contain all these redundancies. Remember that an equivalence class of DAGs can be uniquely represented by a CPDAG, which is precisely what the GES searches for.

The second idea is that by having a decomposable score, we can relatively cheaply check what the score-impact is of adding or removing a single edge and use this to move through the space in a greedy fashion. This means that every possible edge addition (or removal) is evaluated, and then the best action is chosen; instead of choosing an action after each test, as is the case in the PC algorithm. Perhaps surprisingly, Chickering 2000 has shown, by proving the conjecture set out by Meek 1997, that GES (in the limit for large datasets) finds the perfect CPDAG.

The algorithm itself consists of two parts, the forward equivalence search (FES), where edges are greedily added to the graph, followed by the backward equivalence search (BES), where edges are greedily removed from the graph. The flow and main points of the algorithm are as follows.

I. FORWARD EQUIVALENCE SEARCH: The goal of FES is simply to find a CPDAG that is an independence map of the true CPDAG. In theory, we could just take the fully depen-

dent CPDAG, and this entire step could be skipped. However, this would place too much burden on the BES. Ideally the result of FES is a sparse graph that still satisfies the independence map condition. To that end, we start with a fully unconnected graph, and repeat the following three steps.

1. CHECK PROPOSED INSERTS: For each pair of non-adjacent nodes v and w, and for each subset T of the nodes that are undirected adjacent to w and not adjacent to v, we calculate the score improvement of inserting $v \rightarrow w$ and orienting $t—w$ as $t \rightarrow w$ for all $t \in T$. Additionally we check the validity of this change, that is: could we potentially orient all the undirected edges of the resulting PDAG in such a way that it respects the current v-structures and acyclicity. Note that the newly added/oriented edges are construed in such a way that they introduce v-structures into the graph. Since we are searching the space of CPDAGs, v-structures are a focal point of the algorithm.

2. APPLY INSERT: If none of the valid inserts yields a positive improvement we terminate the FES part. Otherwise we apply the insert that promises the greatest improvement.

3. CONVERT TO CPDAG: The resulting PDAG need not be *completed*. To that end, we convert it into a DAG by orienting all undirected edges in an arbitrary way that respects v-structures and acyclicity (note that the validity check in step I ensures that this is possible). From the resulting DAG we can now create the CPDAG.

II. BACKWARD EQUIVALENCE SEARCH: Now that we have an independence map, BES is used to trim the graph down by finding more (conditional) independencies, again in a greedy fashion. Starting with the graph found in the FES part, we repeat the following steps.

1. CHECK PROPOSED DELETIONS: For each pair of adjacent nodes v and w (either directed or undirected), and for each subset H of the nodes that are undirected adjacent to w and adjacent to v, we calculate the score improvement of deleting the edge between v and w, and orienting all undirected edges between $\{v, w\}$ and H, as pointing into H. As in FES, we check the validity of each change.

2. APPLY DELETION: Terminate the BES if no valid positive improvements are found, otherwise, apply the deletion with the greatest improvement.

3. CONVERT TO CPDAG: In the same way as in FES, the possibly non-completed PDAG is converted into a CPDAG.

The resulting graph is a CPDAG that represents an equivalence class of DAGs that all fit the data equally well.

2.3.3 *Fast greedy search*

Ramsey et al. 2017 build on the work of Chickering 2002 by proposing the fast greedy search (FGS) algorithm. The main ideas of this approach are the same as in GES, with four modifications.

The most significant improvement is the parallelization of each of the forward and backward steps. Specifically this concerns point 1. of both the FES and the BES, which require lots of calculations that do not depend on each other.

The second modification is the optional assumption of *single-edge faithfulness*, meaning that a direct edge between two variables will never be added during the FES if they are not correlated

with each other. In some cases this might be incorrect, e.g. perfectly cancelling paths, but it can speed up the search considerably for high-dimensional problems.

The third modification is the explicit use of the Bayesian information criterion (BIC) with a penalty term to score the graphs. The penalty can be adjusted to force sparser graphs. Chickering also mentions the BIC, but does not recommend any specific scoring mechanism in particular. Instead, he gives the properties that a scoring function should adhere to in order to be used for GES, these properties include decomposability and equal scoring for equivalent DAGs.

Their fourth modification is the caching of the calculated score-improvements. Since most of these will be the same in the next step, this saves considerable computation time at the expense of a small amount of extra memory usage. Although Chickering 2002 does not mention this explicitly it seems likely that this was also implied in GES.

As per the title of the work of Ramsey et al. 2017, this approach can handle high-dimensional problems, in the order of a million variables. This would be more than sufficient for our investigation; however, the example problem with 1,000,000 variables by which they show this, has a sample size of only 1,000 observations. Our multimodal single-cell data has sample sizes that are some two orders of magnitude greater; casting serious doubts on the feasibility of applying FGS to our problem.

2.3.4 *Fast causal inference*

Reasoning from the basis of the PC algorithm, a number of problems appear when taking *latent confounders* into consideration. These are unobserved variables that have a causal effect on multiple observed variables. In DMGs they are usually denoted by a bidirectional edge between each pair of observed variables that have a common latent confounder.

The first problem is that the observed variables alone are no longer causally sufficient. Consequently, finding the subset conditioned on which two variables are independent may not be possible if such a subset would have to contain a latent variable. Second, the PC algorithm drastically reduces the number of computations by assuming that any conditioning set is a subset of the adjacent nodes of one of the variables. Now, however, such a variable can also have any number of unobserved parents through which dependencies can slip through.

The fast causal inference (FCI) algorithm, proposed and described in Spirtes, Glymour, and Scheines 2001, addresses these issues while trying to retain some of the computational efficiencies of the PC algorithm. It starts with the same procedures to construct a skeleton and identify v-structures, it is then followed by a correction step to create a trimmed-down skeleton, after which we again identify v-structures and finalize by orienting implied directions.

Note that in contrast to the previously discussed algorithms, FCI assumes that the data was generated by an ADMG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{F} \rangle$ (instead of a DAG). The algorithm builds up a DPAG $\mathcal{H} = \langle \mathcal{V}, \tilde{\mathcal{E}}, \tilde{\mathcal{F}}, \tilde{\mathcal{D}}, \tilde{\mathcal{C}} \rangle$ that should contain $\mathcal{G}$.

I. SKELETON CONSTRUCTION: This step is the same as step I in the PC algorithm, but instead of undirected edges (—), the resulting graph $\mathcal{H}$ is now a skeleton of undirected open-circle edges (∘–∘).

II. V-STRUCTURE IDENTIFICATION: This is also similar to step II of the PC algorithm, with the modification that also bidirectional edges can be formed now.

That is, instead of looking for uncovered triplets of nodes of the form $v_1—v_2—v_3$, we look for uncovered triplets of nodes of the form $v_1 \ast\!\!\circ\!\!\rightarrow v_2 \leftarrow\!\!\circ\!\!\ast v_3$ in $\mathcal{H}$; and, if orientation is warranted, orient only the open-circle edge-ends attached to the middle node into arrowheads.

III. POSSIBLE D-SEPARATION TRIMMING: Here, we will remove any edges from $\mathcal{H}$ that are actually incorrect, but have remained after step I due to latent confounders. This is done by identifying for each node v its *possible d-separation* set $S_v \subseteq \mathcal{V}$, which contains all nodes that can be reached via a path that contains only colliders or nodes that could still be a collider but have not been identified in step II as such because it is part of a triangle. That is, every non-end-node u of such a path is of the form $\ast\!\!\rightarrow u \leftarrow\!\!\ast$, $\ast\!\!\circ\!\!\rightarrow u \leftarrow\!\!\ast$, $\ast\!\!\rightarrow u \leftarrow\!\!\circ\!\!\ast$, or $\ast\!\!\circ\!\!\rightarrow u \leftarrow\!\!\circ\!\!\ast$.

To test for conditional independence between two adjacent variables $v, w \in \mathcal{V}$, it is now sufficient to try all subsets of $S_v \backslash \{v, w\}$ (or of $S_w \backslash \{v, w\}$) for conditional independence, if this is found, the edge can be removed.

IV. V-STRUCTURE RE-IDENTIFICATION: Now we return the resulting graph again into skeleton form by turning all edges into undirected open-circle edges ($\circ\!\!-\!\!\circ$). Once more we perform step II, with the slight modification that if a triplet of nodes $\langle v_1, v_2, v_3 \rangle$ is not identified as a collider, we then mark it as a *definite non-collider*.

V. IMPLIED EDGE ORIENTATION: Similar to the PC algorithm's step III, we now orient any edges that are implied by the orientations already filled in, of course taking care of the existence of bidirected edges.

This is done by *(i)* identifying edges that would otherwise create a cycle; *(ii)* using the identified conditioning sets to find additional colliders; *(iii)* resolving colliders in triangles; and *(iv)* by identifying nodes that would otherwise become colliders but had been marked as definite non-colliders.

As is shown by Zhang 2008, the resulting DPAG of the FCI algorithm is a CDPAG.

2.3.5 *FCI and cyclical causality*

Besides the handling of latent confounders, the FCI algorithm is important because, as Mooij and Claassen 2020 surprisingly showed, it can easily be adapted to represent cyclical causal relationships. In order to do that, we must first acknowledge that the concept of d-separation falls apart if we allow for cycles in a causal graph. Intuitively this is because nodes in the same cycle are too tightly linked together, infinitely causing and being caused by each other, such that d-separating them does not coincide with making them conditionally independent.

To this end, Forré and Mooij 2018 introduce the idea of σ-separation as an extension of the notion of d-separation in possibly cyclical DMGs, defined and discussed in Section 2.2.2. For all subsequent definitions leaning on the notion σ-separation holds that they reduce to the definition of their d-separation counterpart in case of an ADMG.

Forré and Mooij also propose a search algorithm to use their σ-separation concept in a practical setting. However, it can handle only a handful of parameters, so we will not discuss it here, but instead move directly on to the proposal by Mooij and Claassen 2020. Interestingly, this proposal does not involve any modification to the FCI algorithm itself, but instead simply a modification to the interpretation of its input and output.

Input-wise, the FCI algorithm can be fed the (conditional) independences of a possibly cyclical DMG, instead of being limited to ADMGs. On the output-side, the resulting CDPAG can be interpreted to represent possibly cyclical DMGs as described in Section 2.2.4.

Mooij and Claassen 2020 go on to derive various methods to identify properties about the underlying DMGs from the algorithm's output. These include *(i)* identification of non-confounded pairs, *(ii)* identification of direct (non-)causes, *(iii)* identification of non-cycles, and *(iv)* identification of (non-)ancestral relations. In this work we shall focus only on the last of these identification methods, as defined in Section 2.2.4, for the reasons outlined below.

Later we shall look at partially overlapping subsets of variables, where each subset has its own CDPAG. An important question we shall then need to answer is whether the CDPAGs of two such overlapping subsets imply non-contradictory statements about their underlying DMGs. With this in mind, the (non-)ancestry of two variables is a useful property; if $v \in \text{AN}_{\mathcal{G}}(w)$ in the context of one subset, but $v \notin \text{AN}_{\mathcal{G}'}(w)$ in the context of another, then these are clear contradictory statements about the true causal relationships underlying the data.

Contrary to this, the other identification rules have limited utility. Note, for example, that a common cause of two variables in one subset may be a latent confounder in the other. Similarly, causes may be direct in one subset but indirect in the other; or multiple variables may be part of a cycle in one subset but not so in the other. Note that the above identification rule *(iii)* only concerns the identification of definite non-cycles; not the identification of cycles, which would have provided additional useful information on ancestry.

3

DATA

The data used for our analysis comes from a data science problem hosted by Open-Problems 2022 as an open competition (Kaggle 2022). It contains the experimental results produced by each of the two sequencing techniques, CITE and MULTI, discussed in Section 2.1. Both techniques produce two sets of data, one for each of the genetic modalities that they measure, a schematic overview can be seen in Figure 2.

Multimodal Single-Cell Data			
CITE Data (70,988 cells)		MULTI Data (105,942 cells)	
RNA Expression Data (22,050 RNA Genes)	Surface Protein Level Data (140 Proteins)	DNA Accessibility Data (228,942 DNA Genes)	RNA Expression Data (23,418 RNA Genes)
Data Size: 70,988 x 22,050	Data Size: 70,988 x 140	Data Size: 105,942 x 228,942	Data Size: 105,942 x 23,418

Figure 2: Overview of the multimodal single-cell data, split between the two sequencing methods (CITE and MULTI), each with measurements of two modalities of genetic information.

The number of observations (number of cells) is relatively modest for both sequencing methods, but the great number of variables per observation quickly make for substantial datasets. Specifically the chromatin accessibility data, which measures the DNA modality, contains a prohibitive amount of data. One of the main challenges of this work will be to manage these sizable datasets. First, however, we present a preliminary exploration of the data, first for CITE, and then for the MULTI data. Subsequently we will briefly discuss the importance of the RNA modality in combining the data from both techniques, and finally we shall discuss the method used for reducing the dimension of the DNA accessibility data.

3.1 CITE DATASETS

For each cell processed by CITE, we have 22,050 expression measurements of different RNA genes, and 140 surface level measurements of different proteins. In the left panels of Figure 3 one of these cells is visualized. There are 449 RNA genes for which all measured cells report an expression of 0, these genes are left out of our investigation, leaving a total of 21,601 RNA genes remaining.

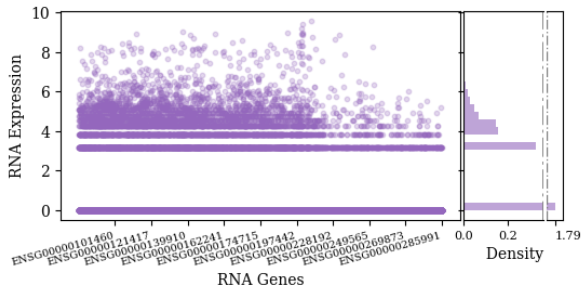
The RNA genes are named for their Ensembl gene ID (Ensembl 2023), and calculated by transforming their measured occurrence count to the natural logarithm of 1 plus its library-size normalized value. That is, each raw count $f_{g,c}$ of gene g and cell c is transformed to:

$$f'_{g,c} = \log\left(1 + \frac{k}{\sum_{g'} f_{g',c}} \cdot f_{g,c}\right) \tag{1}$$

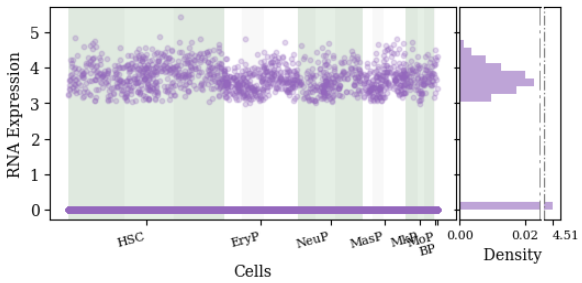
where k is a constant across all cells, back-calculation shows that this value was set at k=1,000,000. Because all the counts from the same cell are scaled by the same factor, we observe the regular pattern in Figure 3a. Also note that we do not observe a similar pattern when we look at a single RNA gene across different cells, as in the Figure 3b, since each data-point is transformed by a different, cell-specific, scalar.

The protein data is the most modest in terms of dimensionality, with only 140 different proteins for which measurements have been taken. Partly this is because there are a bit fewer human proteins than there are genes; partly because CITE measures only proteins on the cell surface; but mostly this is because, for every protein that could potentially be measured by this technique, a specific ADT needs to be identified and added to the process. This inherent form of physical whitelisting rather limits the scope of proteins that can be measured.

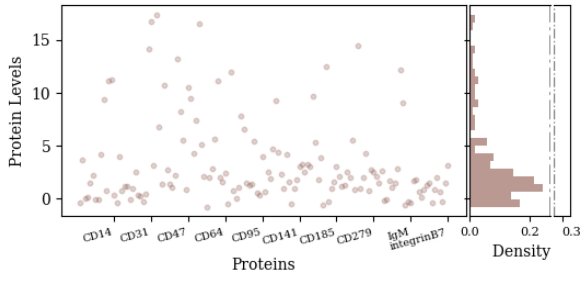
The protein level measurements are denoised and scaled by background (DSB), this is a normalization method described by Kotliarov et al. 2020 and designed for CITE data specifically. It uses droplets that are known to contain very little RNA, and as such can be used to reliably measure background protein levels that can act as a benchmark from which to measure the protein levels during CITE. Note that because both the background levels and the actual levels are estimated values, the DSB-normalized levels can be negative, as is visible in Figure 3c.



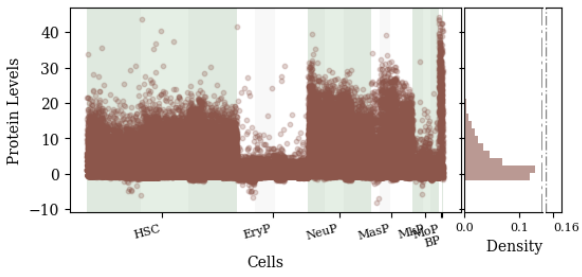
(a) RNA expression for cell `#035884` (n: 21,601; non-zero: 32%).



(b) Expression of RNA gene `ENSG00000239779` (n: 70,988; non-zero: 2.1%).

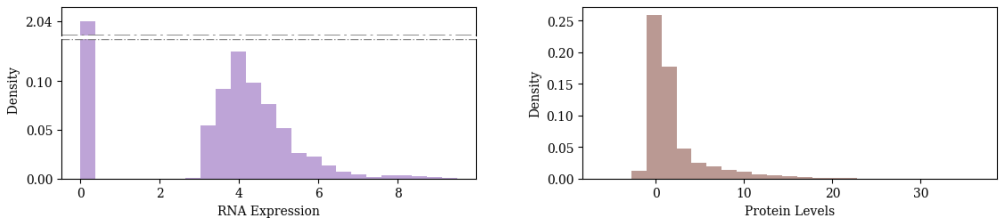


(c) Protein levels for cell `#035884` (n: 140; non-zero: 100%).



(d) Levels of protein `CD48` (n: 70,988; non-zero: 100%).

Figure 3: Example slices of data from the CITE datasets of the RNA modality (top) and protein modality (bottom), taken from a single cell across genes and proteins (left) or across a single gene and a single protein across all cells (right). The alternating green areas in (b) and (d) indicate different cell types and the alternating shades within them indicate distinct test-days. All values are represented in their normalized form.



(a) log1p normalized RNA expression counts (n: 1,533,411,788; non-zero: 22%).

(b) DSB normalized surface protein levels (n: 9,938,320; non-zero: 100%).

Figure 4: Distribution of CITE data across all RNA genes (a), all proteins (b), and all cells (both).

The right panels of Figure 3 show the CITE data from another perspective, namely a single RNA gene and a single protein across all measured cells. As an additional piece of context, all the cells in the datasets have been annotated with a cell-type, which the publishers of the data have pre-estimated based on the work of Velten et al. 2017. To be clear, this extra information is an estimate based on the RNA data and it is not a ground truth. It will also not play a further role in our work, but it is nevertheless worth mentioning, as the figure shows that especially the protein levels seem distinctly correlated with the type of cell that they are in. Sampling from the donors took place over three different test-days which are also marked in the right panels. Any dependence on this is less apparent from the graphs, though some correspondence seems visible from Figure 3d.

From the distribution histograms in Figure 4 of the full datasets we glimse that the RNA expression data is relatively sparse, with only 22% of the data-points being greater than zero. The protein level data is measured on a much more continuous spectrum and as such displays no such sparsity.

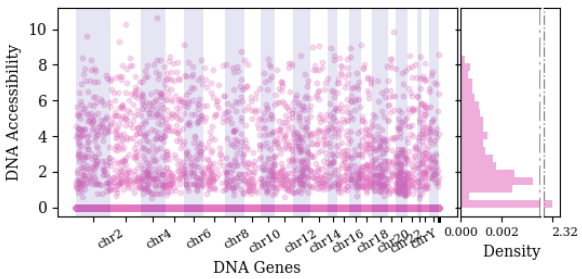
3.2 MULTI DATASETS

The MULTI datasets contain observations for more cells than the CITE datasets, this is mainly because there was one additional test-day, four instead of three. For each of these cells we have measurements for 228,942 DNA genes, this already makes the MULTI data many times more voluminous then the CITE data. All these cells also have measurements for 23,418 RNA genes, but just as with the CITE data, these include several genes for which all measurements are 0. Eliminating these from our dataset leaves a total of 22,858 RNA genes.

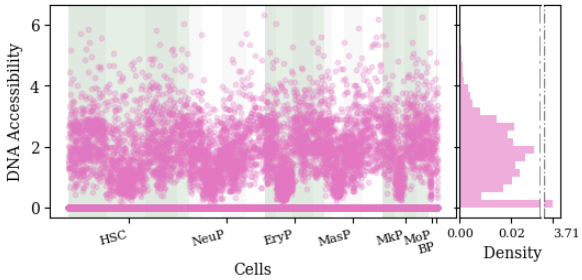
The DNA genes are labeled by their chromosome and their genomic coordinates on that chromosome on the reference genome specified by 10xGenomics (2020). In Figure 5 the two modalities of an example cell are visualized. The RNA expression depicted in panel 5c should look familiar, it is the same kind of data and transformed in the same way as the CITE RNA data we saw in Figure 3a.

The DNA data is calculated by applying the term frequency inverse document frequency (TF-IDF) normalization. This is a transformation routinely used for natural language models, but it is also commonly used for chromatin accessibility peak counts (e.g. Cusanovich et al. 2018). It consist of two parts, the term frequency $\mathrm{tf}_{g,c}$ of gene g and cell c is calculated as the normalized occurrence of g in c:

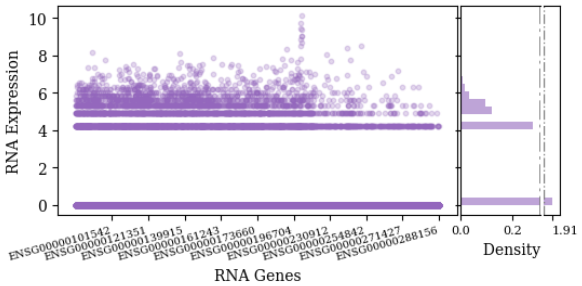
$$\mathrm{tf}_{g,c} = \frac{\mathrm{f}_{g,c}}{\sum_{g'} \mathrm{f}_{g',c}} \tag{2}$$



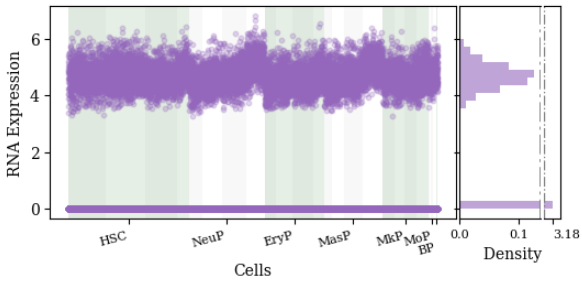
(a) DNA accessibility for cell `#010452` (n: 228,942; non-zero: 0.98%).



(b) Accessibility of DNA gene `chr6:149450686-149451557` (n: 105,942; non-zero: 6.2%).



(c) RNA expression for cell `#010452` (n: 22,858; non-zero: 23%).



(d) Expression of RNA gene `ENSG00000239779` (n: 105,942; non-zero: 14%).

Figure 5: Example slices of data from the MULTI datasets of the DNA modality (top) and RNA modality (bottom), taken from a single cell across DNA and RNA genes (left) or across a single DNA gene and a single RNA gene across all cells (right). The alternating blue areas in (a) indicate different chromosomes. The alternating green areas in (b) and (d) indicate different cell types and the alternating shades within them indicate distinct test-days. All values are represented in their normalized form.

where $f_{g,c}$ is the raw peak count of gene g in cell c. The inverse document frequency part, idf_g, is the inverse fraction of all cells that contain gene g:

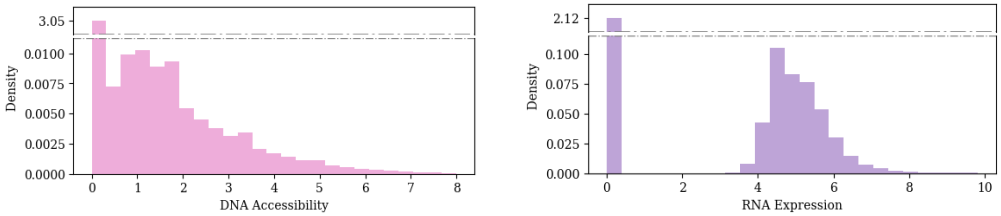
$$\mathrm{idf}_g = \frac{\sum_{c'} 1}{\sum_{c'} [f_{g,c'} > 0]} \tag{3}$$

using the Iverson bracket notation. Combining the above two factors translates the raw chromatin accessibility peak counts to their normalized variants as follows:

$$f''_{g,c} = \log(\mathrm{tf}_{g,c}) \cdot \log(\mathrm{idf}_g). \tag{4}$$

The right panels of Figure 5 show examples of the two genetic modalities across all cells. Again they are ordered first by cell-type, and then by test-day. Besides a correlation to cell-type, we can also speculate as to a correlation to test-day, both for the DNA data and the RNA data this example seems to indicate such a dependence.

Moving beyond examples, Figure 6 displays the distribution of normalized measurements from both modalities. Similar as in the CITE datasets, the RNA data is quite sparse, but not as sparse as the DNA data. With over $24 \cdot 10^9$ data-points this is by far the largest dataset, however some 97.5% of those values are zero.



(a) TF-IDF norm. DNA chrom. accessibility (n: 24,254,573,364; non-zero: 2.5%).

(b) log1p normalized RNA expression counts (n: 2,421,622,236; non-zero: 17%).

Figure 6: Distribution of MULTI data across all DNA genes (a), all RNA genes (b), and all cells (both).

3.3 REDUCING DNA DIMENSIONALITY

The largest dataset under our consideration concerns the DNA modality of the MULTI method. With 228,942 variables per observation it is too large to feasibly include all these variables as nodes in a causal graph. Consequently, before using this data in any causal inference algorithm, we should decrease the dimensionality of this particular modality.

Among the methods considered for this job are linear dimension-reduction techniques like the principal component analysis (PCA) (Pearson 1901), and singular value decomposition (SVD) (Eckart and Young 1936), as well as non-linear approaches like autoencoders (Kramer 1991). However, none of these methods managed to significantly reduce dimension while retaining a substantial fraction of the information in the DNA data. This is why ultimately we settled on a non-standard method that is more specific to the data at hand, and at least retains a certain degree of interpretability for the resulting variables.

We do this by taking into consideration some additional information that we have on each DNA gene in the dataset. Namely, we know on which chromosome the gene lies, and what its start- and end-location is on that chromosome: its genomic coordinates. To view this in its context, again consider Figure 5a where the DNA data is ordered by chromosome and genomic coordinates across the x-axis, or consider Figure 5b that displays the data of the DNA gene on chromosome 6 between genomic coordinates 149,450,686 and 149,451,557.

We divide each chromosome into segments that are roughly evenly large in terms of genomic coordinate-distance. Concretely, we set a target size s of 10,000,000 and define, for chromosome c, the number of segments N_c, and the length of each individual segment L_c respectively as:

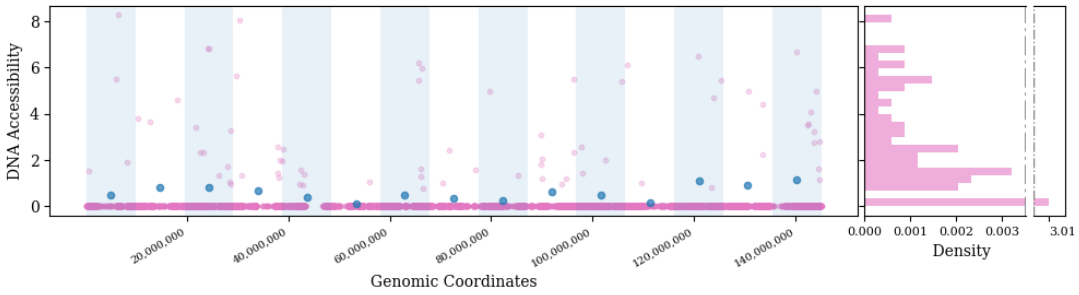
$$N_c = \left\lceil \frac{i_{c,max} - i_{c,min} + 1}{s} \right\rceil \qquad \text{and} \qquad L_c = \frac{i_{c,max} - i_{c,min} + 1}{N_c}. \tag{5}$$

Here, $i_{c,min}$ and $i_{c,max}$ are the lowest and highest genomic coordinates respectively on chromosome c present in the data. Each chromosome c is now subdivided into N_c segments spanning a genomic coordinate distance of L_c, and every DNA gene in the data is assigned to the segment that covers its middle coordinate. Let $A_{c,k}$ be the set of all DNA accessibility values of genes in the k'th segment of chromosome c. These values are aggregated into a single value by way of the following calculation:

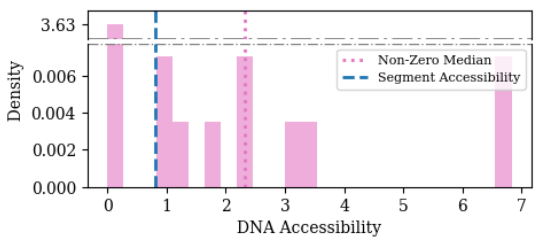
$$\overline{a}_{c,k} = \text{median}\{a \mid a \neq 0, a \in A_{c,k}\} \cdot \frac{\log\left(1 + \sum_{a \in A_{c,k}} [a \neq 0]\right)}{\log\left(1 + \sum_{a \in A_{c,k}} 1\right)}, \tag{6}$$

such that the aggregated accessibility value of a segment consists of two factors, the median of non-zero values, scaled by a logarithmic measure of the presence of non-zero values.

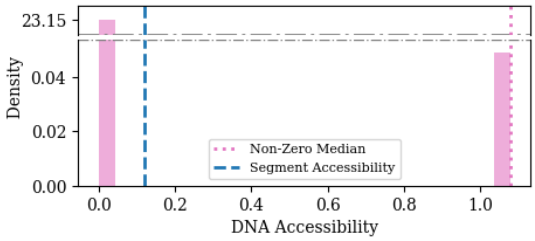
Figure 7 illustrates how this works out for a single example cell on chromosome 8, the 10,361 dimensions of the full DNA chromatin accessibility data for this chromosome have been reduced to a mere 15. The bottom panels further clarify how the two factors tackle the high degree of sparsity of the data, on the left we see that the segment accessibility stays relatively close to the non-zero median because of the decent amount of non-zero values in the segment; whereas, on the right, the segment accessibility is almost reduced to zero.



(a) DNA accessibility on chromosome `chr8` (n: 10,361; non-zero: 0.7%).



(b) DNA accessibility on segment `#02` of `chr8` (n: 1,039; non-zero: 0.96%).



(c) DNA accessibility on segment `#05` of `chr8` (n: 472; non-zero: 0.21%).

Figure 7: Example of the DNA chromatin accessibility for MULTI cell `#010452` of a single chromosome, with the corresponding aggregated segment accessibility values (top). The alternating blue areas indicate different segments on the chromosome. The bottom panels each show, for a single segment (0-indexed), how the aggregated value relates to the data-points within that segment.

It should be noted that it is a bit awkward to squeeze the two factors that make up $\overline{a}_{c,k}$ into a single value. An obvious remedy to this would be to let every segment be represented by a two-dimensional value. However, performing causal inference on multi-dimensional nodes comes with a lot of complications that are beyond the scope of this work; which is why we have opted for this single-value-approach instead.

In Figure 8 the results of this compression method are visualized. We see that the higher-numbered chromosomes have less segments, as they are shorter in length. Note also that besides the 22 numbered chromosomes, and the X- and Y-chromosome, there are also some rest-categories (named with the prefix `GL` or `KI`) with genes whose location on the genome

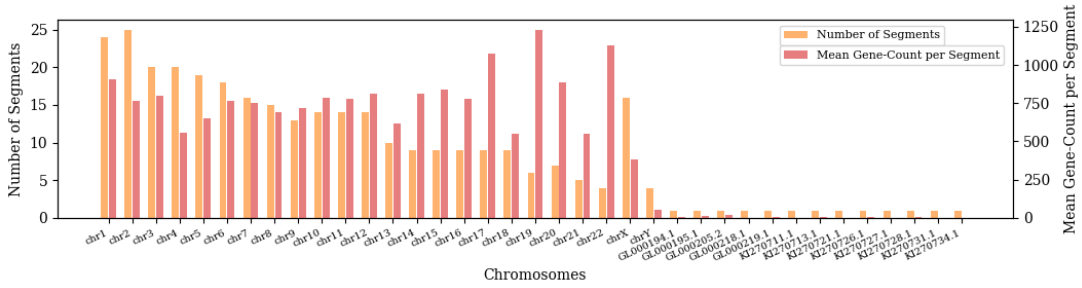
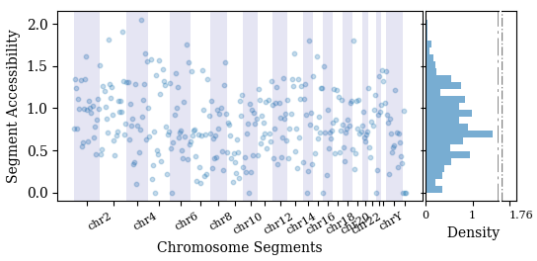


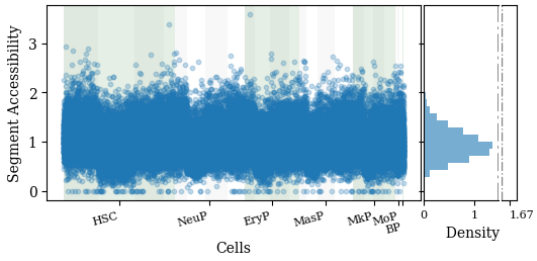
Figure 8: Number of segments per chromosome, and corresponding mean gene-count per segment.

is unknown. These all contain very few genes and are discarded from consideration going forward.

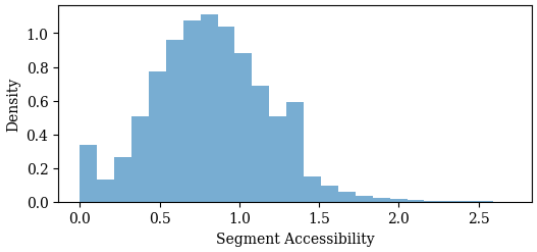
In total this procedure reduces the dimension of the data substantially from 228,942 variables to 309. The final result is visualized in Figure 9, which can be seen as an extension of Figure 5 and Figure 6. The compressed segment data is shown across all segments for a single cell, and across all cells for a single segment.



(a) Segment agg. accessibility for cell `#010452` (n: 309; non-zero: 97%).



(b) Agg. accessibility of segment `chr8_02` (n: 105,942; non-zero: 100%).



(c) Segment aggregated DNA chrom. accessibility (n: 32,736,078; non-zero: 97%).

Figure 9: Dimension-reduced MULTI DNA dataset by way of chromosome segment aggregation. Top: example slices of data taken from a single cell across segments (left), or from a single segment across all cells (right). Bottom: distribution of segment accessibility across all segments and all cells. The alternating blue areas in (a) indicate different chromosomes.The alternating green areas in (b) indicate different cell types and the alternating shades within them indicate distinct test-days.

3.4 SUMMARY

Figure 10 shows an overview of the resulting data after the steps described in the previous sections. Comparing it to the raw data summarized in Figure 2 the data is reduced in size: some of the RNA data has been discarded for both CITE and MULTI due to 0-variance across

all cells; and the high-dimensional DNA chromatin accessibility data has been replaced by its aggregated values per chromosome segment. In subsequent chapters when we mention DNA data, it will be these aggregated values that we refer to.

An important final note on the data is that both the CITE and MULTI techniques measure RNA expression, consequently this is the modality that will ultimately have to link the causal relationships together. However, the two RNA datasets themselves have less overlap than might be expected. In total there are 18,021 RNA genes that occur in both techniques' final selection of data. This seems enough to create a bridge between the two, however it does mean that there are 3,580 RNA genes that occur only in CITE, and 4,837 RNA genes that occur only in the MULTI data.

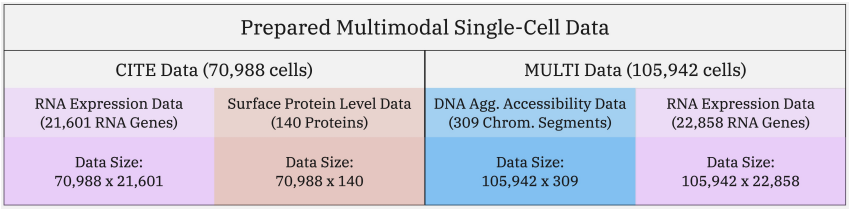


Figure 10: Overview of the multimodal single-cell data after the application of all data preparations.

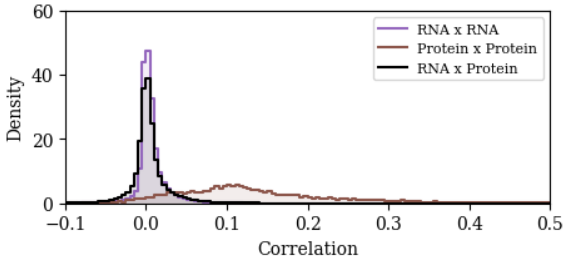
4

METHODS

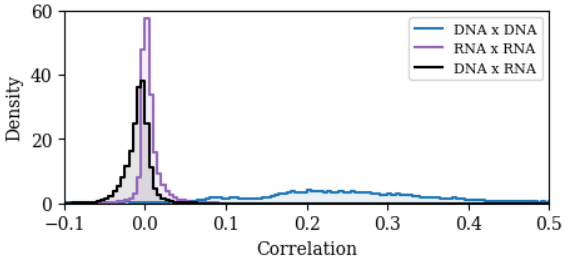
The approach that we propose to deal with high-dimensionality and cyclical causality is comprised of two parts that are both described in this chapter. First we discuss the subdivision of the variable-spaces into small, partially overlapping clusters; next we discuss the execution of the FCI algorithm on each of these individual clusters.

4.1 CLUSTERING

After the measures taken in the previous chapter, the CITE and MULTI datasets still contain 21,741 and 23,167 variables respectively; orders of magnitude more than can be handled by cyclical causal inference algorithms. In order to make this task manageable, we first perform a clustering procedure which produces small clusters of variables that slightly overlap with each other. This overlap can later be used to verify the consistency of causal relationships that we find in individual clusters, and future research could potentially use such overlap to generate a single aggregate causal network from all these sub-networks.



(a) Correlations of CITE data among RNA genes (n: 233,290,800), among proteins (n: 9,730), and between RNA genes and proteins (n: 3,024,140).



(b) Correlations of MULTI data among DNA chrom. segments (n: 47,586), among RNA genes (n: 261,232,653), and between DNA chrom. segments and RNA genes (n: 7,063,122).

Figure 11: Distribution of correlations of (a) the CITE data (RNA expression and protein levels), and (b) the MULTI data (chromosome segment DNA accessibility and RNA expression).

The basis of the used clustering procedure is the Spearman's rank correlation, which is derived by first transforming the values of each variable to their corresponding rank values and calculating the Pearson correlation between those. This choice was made in order to accommodate the sparse and asymmetric nature of our data. Alternatively we could have assumed linearity, or we could have employed estimators capable of discerning more general dependencies. However, using linear estimates, like directly applying Pearson correlation to the data, does not seem to be justified, see: Figure 3, Figure 5, and Figure 9; and the use of more general

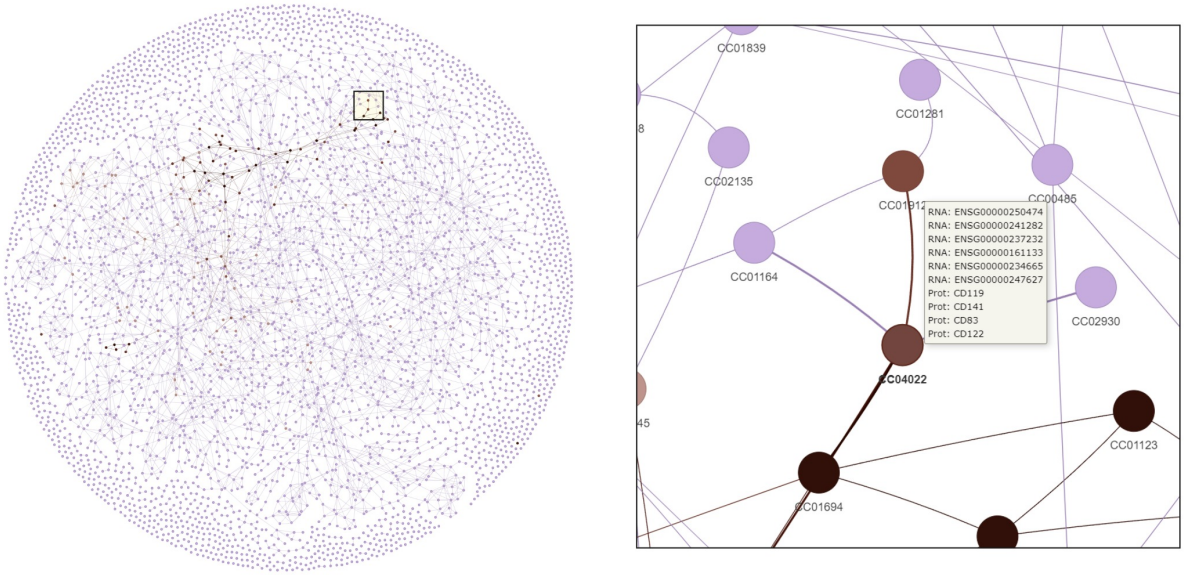
dependence-tests, like calculating *mutual information* (Shannon 1948) on continuous variables, are computationally too intensive to be feasible for our large datasets.

Figure 11 visualizes the calculated correlations by splitting them up by paired modalities. Correlations among RNA genes tend to be quite close to zero and are slightly positively biased (owing at least in part to the sparsity of these variables), whereas among the denser protein data, and the denser DNA (chrom. segment) data, we observe more pronounced positive correlations. Interestingly, while the correlations between RNA and proteins tend to be positive, in line with the primary biological relationship between the two; correlations between RNA expression and DNA accessibility tend to be negative. The latter perhaps pointing to the (indirect) inhibitory effects of RNA on DNA outpacing the promotion by DNA of RNA in terms of how many different genes are involved in either process.

Given these correlations, all the variables $v \in \mathcal{V}$ are assigned to (possibly multiple) clusters of size s. This is done with an iterative procedure during which we keep track of the counter $n(v)$, which is the number of clusters that variable v is already assigned to, and we start by creating a new cluster $\mathcal{C}$ that contains an arbitrary variable v_1 for which $n(v_1)=0$, that is: $\mathcal{C} = \{v_1\}$. Subsequently, the next step is repeated $s-1$ times in order to create a cluster of size s:

$$\mathcal{C} := \mathcal{C} \cup \left\{ v \;\middle|\; v = \underset{v' \in \mathcal{V} \setminus \mathcal{C}}{\arg\max} \left\{ \sum_{w \in \mathcal{C}} \lambda^{n(v')} \cdot \lambda^{n(w)} \cdot \left| \rho(v', w) \right| \right\} \right\}. \tag{7}$$

Here, $\rho(v', w)$ denotes the Spearman's rank correlation between the variables v' and w, and $\lambda \in (0, 1]$ is a decay factor that diminishes the correlations to make it less likely that correlations of variables that are already included in other clusters dictate the selection of new variables.



(a) Full overview

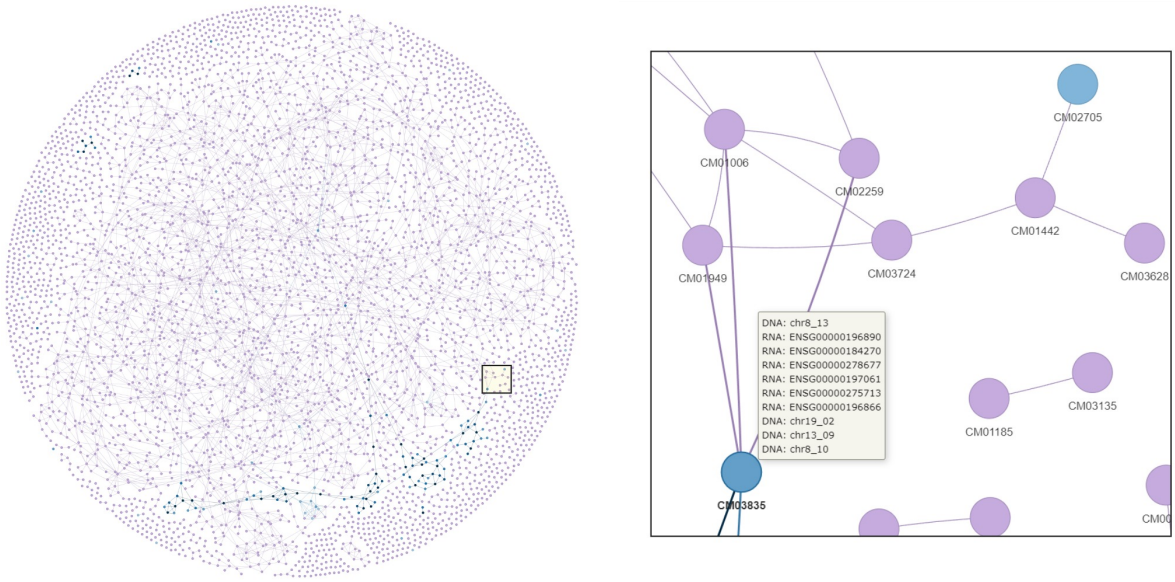
(b) Inset highlighted in (a), cluster `CC04022` is shown to contain 6 RNA variables and 4 protein variables

Figure 12: Clustering of CITE data, each node (n: 4,181) represents a single cluster, each edge (n: 4,694) indicates that the two adjacent clusters share at least two variables. Clusters containing RNA variables exclusively are shown in purple, clusters that also contain protein variables are shown in red (darker shades of red indicate higher concentrations of protein variables).

We can describe Equation 7 as adding to the cluster a new variable that has the highest average absolute decayed correlation with the cluster's other variables. This process of creating clusters is then repeated until there are no variables v left with $n(v)=0$. In other words: we stop creating new clusters if every variable $v \in V$ belongs to at least one cluster.

The cluster size is set relatively low at $s=10$ in order to facilitate the execution of the computationally heavy cyclical causal inference algorithms later on. The decay factor λ influences to what degree the same variable is chosen again and again for multiple clusters, as such it determines how many clusters are needed in total to cover all variables. Ideally, the overlap of clusters is evenly spread out over the space of variables, this argues for a strong aversion to choosing previously chosen nodes, and thus for a low λ. On the other hand, we have to choose previously chosen nodes in order to get the overlap we need to decently connect the space of variables, which argues for a high λ. This trade-off is made by aiming at having every variable on average included in two different clusters. By examining multiple possible values, we settled on $\lambda = 0.7$, which approximately satisfies this aim for both datasets. With this decay factor, a CITE variable is on average included in 1.92 clusters, and a MULTI variable in 1.94.

In Figure 12 the results of this clustering method applied to the CITE data is visualized. The 21,741 variables of this dataset are assigned to a total of 4,181 clusters, which are represented as nodes in the displayed network. Clusters that share two or more variables carry special significance because this will later allow us to verify whether the causal networks found for both clusters are consistent with each other. As such, clusters that have this overlap are joined together by an edge in the visualization. We can see in the figure that the protein variables tend to cluster together, consistent with the higher correlations we saw in Figure 11a.



(a) Full overview

(b) Inset highlighted in (a), cluster `CM03835` is shown to contain 6 RNA variables and 4 DNA variables

Figure 13: Clustering of MULTI data, each node (n: 4,497) represents a single cluster, each edge (n: 5,328) indicates that the two adjacent clusters share at least two variables. Clusters containing RNA variables exclusively are shown in purple, clusters that also contain DNA variables are shown in blue (darker shades of blue indicate higher concentrations of DNA variables).

The same visualization can be seen in Figure 13 for the MULTI data, where the 23,167 variables are assigned to 4,497 clusters. Even more so than for the CITE data, we can observe the

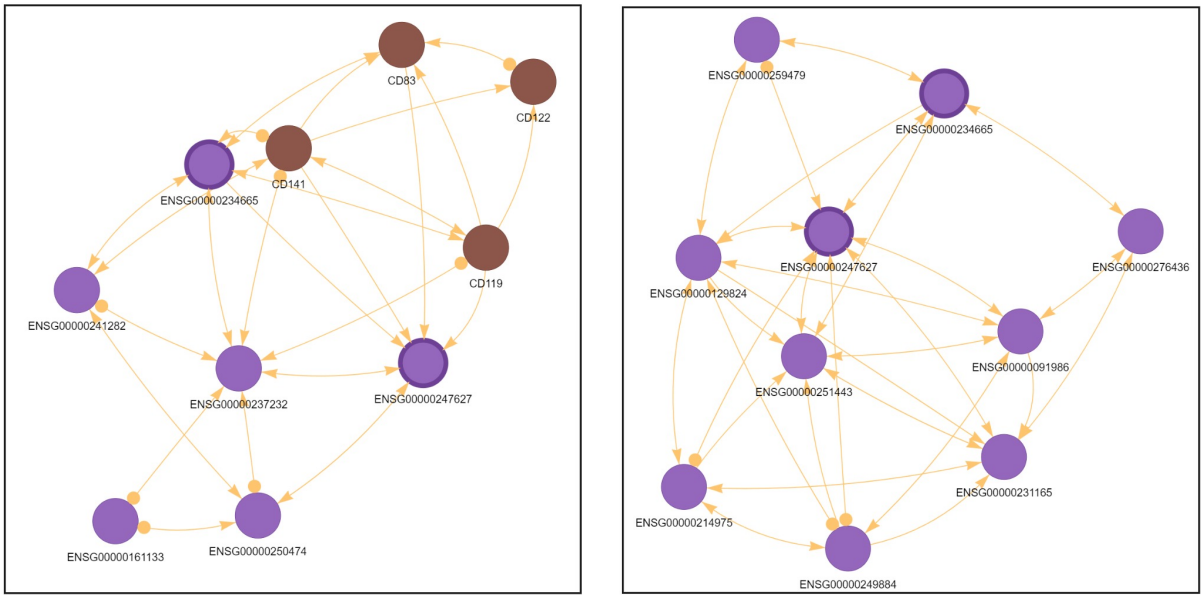
clustering together of non-RNA variables, in line with the even higher correlations observed in Figure 11b for DNA variables among themselves.

4.2 FCI FOR CYCLICAL CAUSALITY

Now that we have the different clusters, we can finally apply causal inference on the data. Each cluster has a mere 10 variables, that number is modest enough to be fed to the FCI algorithm. As discussed in Section 2.3.5, the output of this algorithm can be interpreted as representing a system including cyclical causal relationships, this is the primary reason we employ it in this work.

Similar to the use of Spearman's rank correlation in Section 4.1, and for the same reasons, we run the FCI algorithm on the rank values of the data instead of directly on the data itself. Fisher's Z tests (Fisher 1921) with a standard significance level of 0.05 are used in the FCI runs to test for (conditional) independence.

In Figure 14 two clusters are visualized as examples of the output that this FCI procedure produces. Remember that this output is a DPAG $\mathcal{P}$, where adjacency between two nodes implies the existence of an inducing path between them in any DMG $\mathcal{G}$ that is represented by $\mathcal{P}$. More precisely, an edge $v \to w$ in $\mathcal{P}$ means that v is an ancestor of w in $\mathcal{G}$; $v \leftrightarrow w$ means that v is not an ancestor of w nor vice versa in $\mathcal{G}$; and open-circle edge-endpoints are interpreted as possibly being either an arrowtail or an arrowhead.



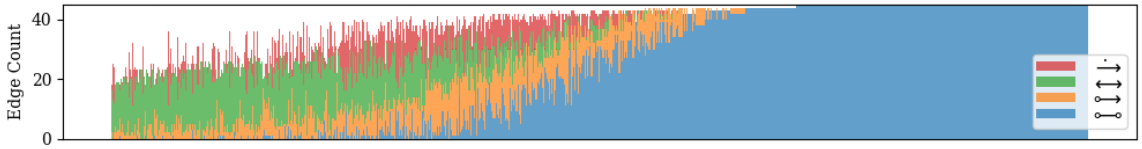
(a) FCI output of cluster `CC04022`.

(b) FCI output of cluster `CC01164`.

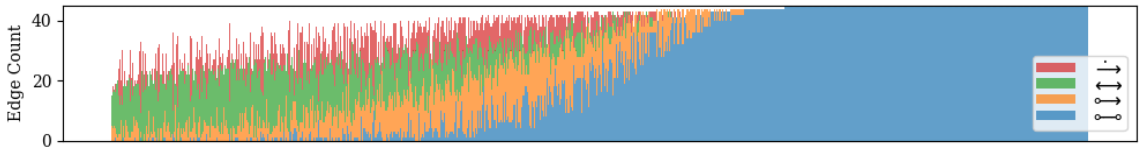
Figure 14: Output of the FCI algorithm on two CITE clusters (open-circle endpoints are represented by filled dots). The two displayed clusters share two variables which are highlighted in both figures (this overlap can also be seen in Figure 12b).

The FCI outputs of both example clusters imply an inducing path between the RNA variables `ENGS00000234665` and `ENGS00000247627`, but they do not agree on the ancestral relationship between the two. Figure 14a clearly designates the former as an ancestor of the latter by way of the directed edge ($\to$) between them, but in Figure 14b we observe no such edge. Stronger still, the graph does not contain a possibly directed path between the two nodes; specifically implying non-ancestry.

To get an idea of what a typical output DPAG looks like beyond these two examples, we turn our attention to Figure 15, here the composition of edge types for each cluster's DPAG is visualized. For both the CITE clusters and the MULTI clusters, the distribution of edge types is very similar. Note that since a cluster has exactly 10 variables, it contains $\frac{1}{2} \cdot 10 \cdot (10-1) = 45$ possible edges, so we can observe that about a third of all clusters yielded a fully connected graph consisting only of undirected open-circle edges ($\circ\!\!-\!\!\circ$). On the other side of the spectrum, we see that there is not a single cluster that contains less than 15 edges in total, which makes sense considering that the clusters were formed by searching for correlated variables, making it highly unlikely that within a cluster many variables would be independent of each other.



(a) CITE clusters (n: 4,181).



(b) MULTI clusters (n: 4,497).

Figure 15: Counts of different edge types resulting from the execution of FCI across all clusters (along the horizontal axis) for both the CITE and MULTI data.

5

RESULTS AND DISCUSSION

With the two-tiered approach to causal inference wrapped up, we now inspect the results. This is done by checking the consistency of ancestral relationships between overlapping clusters, which we shall discuss first. After that, we go over some of the shortcomings of this approach and cover subsequent suggestions for future research.

5.1 CONSISTENCY

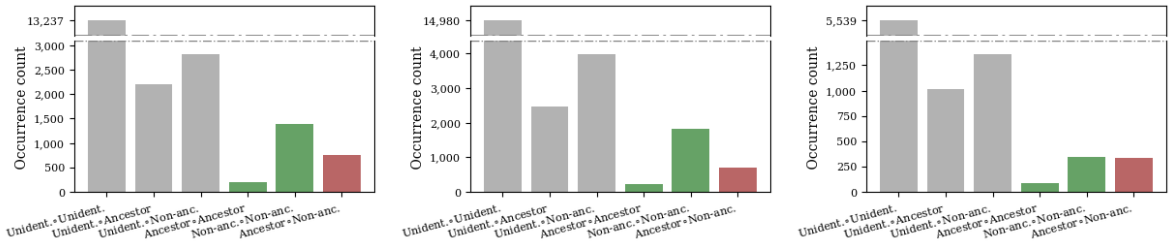
In Section 2.2.4 we ended on the sufficient conditions needed to determine both ancestry and non-ancestry of an ordered pair of distinct variables (v, w). Here we use this to verify the consistency of the output of the FCI algorithm for all the different sub-clusters. In order to do so, we first gather every (unordered) pair of clusters that have an overlap of at least two variables, and subsequently gather every ordered pair of variables within that overlap. This collection of pairs, which we shall call $\mathcal{O}$, contains 20,606 and 24,174 pairs for CITE and MULTI respectively. Additionally, we can look at the overlap between CITE clusters and MULTI clusters. Both clusters' FCI outputs were created with altogether different datasets, but these datasets still share a considerable portion of their RNA variables, leading to 18,906 overlapping pairs in $\mathcal{O}$ for this cross, denoted as CITE$\times$MULTI.

An element $p \in \mathcal{O}$ is a tuple of two variables and an unordered pair of associated clusters: $p = \langle v, w, \{C_a, C_b\}\rangle$. For each of these we can now determine, for both its clusters, what the *verdict* of the FCI's output CDPAG $\mathcal{P}$ is. That is, whether $\mathcal{P}$ either *(i)* implies ancestry $v \in \text{AN}_{\mathcal{G}}(w)$, *(ii)* implies non-ancestry $v \notin \text{AN}_{\mathcal{G}}(w)$, or *(iii)* could not make a statement about v's ancestry of w in the underlying DMG $\mathcal{G}$. In the first two cases, we call the verdict *identified* within the context of that cluster, otherwise we call it *unidentified*. If two verdicts are both identified, but not in agreement with each other, then they are called *inconsistent*, otherwise *consistent*.

Let $X_a(p)$ and $X_b(p)$ be the verdicts on ancestry of $p \in \mathcal{O}$ in the context of its two clusters respectively, and $X_*(p)$ may refer to either of the two. Each of these can take on one of the following three values: (A)ncestor, (N)on-ancestor, or (U)nidentified. Figure 16 visualizes the results of these evaluations, where we can see that the majority of cluster-specific evaluations of an overlapping ordered pair leave its ancestry unidentified. This is important to note, because any duo of verdicts that includes an unidentified one is always consistent.

In combination with the high correlation of the identifiability of paired verdicts, this creates a problem. Namely, if we were to test whether paired verdicts are more often consistent than chance; that is, for independent random pairs p and p':

$$\mathrm{P}\big(X_a(p) \text{ and } X_b(p) \text{ are consistent}\big) > \mathrm{P}\big(X_*(p) \text{ and } X_*(p') \text{ are consistent}\big), \tag{8}$$



(a) CITE (n: 20,606). (b) MULTI (n: 24,174). (c) CITE×MULTI (n: 18,906).

Figure 16: Counts of ordered pairs of distinct variables (v, w) that are together contained in an overlap between two (unordered) clusters; categorized by whether v is an ancestor of w according to the FCI's output CDPAG of both clusters. The evaluation of ancestry is either identifiably yes (*Ancestor*), identifiably no (*Non-anc.*), or unidentifiable from the CDPAG (*Unident.*). The two clusters are either based on CITE data (a), on MULTI data (b), or on one of each (c).

then we would be likely to observe the opposite instead. This is due to the bias of an identified verdict to pair up with another identified verdict, thereby creating the necessary conditions for inconsistency; whereas a random draw would pair most identified verdicts with unidentified ones, never creating these necessary conditions in the first place. To correct for this, we shall leave out any variable-pair for which either cluster claims unidentifiability.

First, however, we should back up the claim of dependence between the identifiability of a pair's two verdicts. To that end, we set up a binomial test with a null-hypothesis that claims that the probability of identifiability is not affected by conditioning on the other cluster's identifiability. That is, for a random pair p:

$$P\big(X_b(p) \neq U \,\big|\, X_a(p) \neq U\big) \;=\; P\big(X_*(p) \neq U\big). \tag{9}$$

For our test, we are interested in the degree to which identifiable ancestries match up in their overlap compared to a random draw from the unconditional verdicts. With that justification, we can use the following estimate of $P(X_*(p) \neq U)$ as the H_0-probability in our test:

$$p_0 \;=\; \frac{1}{2|\mathcal{O}|} \cdot \Big(\sum_{p \in \mathcal{O}} \big[X_a(p) \neq U\big] + \sum_{p \in \mathcal{O}} \big[X_b(p) \neq U\big] \Big). \tag{10}$$

The total count and success count for the test are then calculated by considering every single verdict (two for every pair in $\mathcal{O}$) and check whether this verdict was *(i)* identified for the total count, and *(ii)* identified and paired with another identified verdict, for the success count:

$$\begin{aligned} N_{\text{total}} \quad &= \quad \sum_{p \in \mathcal{O}} \big[X_a(p) \neq U\big] + \sum_{p \in \mathcal{O}} \big[X_b(p) \neq U\big], \\ N_{\text{success}} \quad &= \; 2 \cdot \sum_{p \in \mathcal{O}} \big[X_b(p) \neq U \text{ and } X_a(p) \neq U\big]. \end{aligned} \tag{11}$$

The alternative hypothesis is that the conditional probability from Equation 9 is instead greater. Table 1 shows that the test-statistics, the success-ratios estimated from the counts, are significantly higher than their respective p_0's; leading to minuscule p-values and a sound rejection

DATASET	N_{TOTAL}	$N_{SUCCESS}$	H_0 PROB.	STATISTIC	P-VALUE
CITE	9,728	4,718	0.236	0.485	$0.0 \cdot 10^{-32}$
MULTI	11,941	5,494	0.247	0.460	$0.0 \cdot 10^{-32}$
CITE×MULTI	7,618	2,736	0.201	0.359	$0.0 \cdot 10^{-32}$

Table 1: Results of the binomial tests that test the dependence between the *identifiability* of the two verdicts of a pair $p \in \mathcal{O}$; for both datasets and their cross. The null hypothesis claims the equality of Equation 9, the alternative claims the left-hand side to be greater.

of the null-hypotheses. We are thus justified to exclude the pairs $p \in \mathcal{O}$ which include an unidentified verdict from consideration in our upcoming tests.

With this confirmation in hand, we can now test the consistency of pairs $p \in \mathcal{O}$ with solely identifiable ancestry verdicts. Our null-hypothesis claims that the probability of a pair's verdicts' consistency, given that both are identified, equals the probability that two randomly drawn identified verdicts are consistent. It is captured by the following equation for independent random pairs p and p':

$$\begin{aligned} & P\big(X_a(p) = X_b(p) \;\big|\; X_a(p) \neq U \text{ and } X_b(p) \neq U\big) \\ = \; & P\big(X_*(p) = X_*(p') \;\big|\; X_*(p) \neq U \text{ and } X_*(p') \neq U\big), \end{aligned} \tag{12}$$

with the alternative hypothesis claiming the left-hand side to be greater. The H_0 probability is calculated as the probability of drawing two consecutive A-verdicts or two consecutive N-verdicts from the pool of observed single verdicts that are either A or N:

$$\begin{aligned} n_x &= \sum_{p \in \mathcal{O}} \big[X_a(p) = x\big] + \sum_{p \in \mathcal{O}} \big[X_b(p) = x\big], \quad \text{for} \quad x \in \{A, N, U\}, \\ p_0 &= \Big(\frac{n_A}{n_A + n_N}\Big)^2 + \Big(\frac{n_N}{n_A + n_N}\Big)^2. \end{aligned} \tag{13}$$

For the total count and success count of the test, we consider all pairs (instead of single verdicts) and check *(i)* whether both verdicts of the pair are identified for the total count, and *(ii)* whether, additionally, they are also consistent for the success count:

$$\begin{aligned} N_{\text{total}} &= \sum_{p \in \mathcal{O}} \big[X_a(p) \neq U \text{ and } X_b(p) \neq U\big], \\ N_{\text{success}} &= \sum_{p \in \mathcal{O}} \big[X_a(p) \neq U \text{ and } X_b(p) \neq U \text{ and } X_a(p) = X_b(p)\big]. \end{aligned} \tag{14}$$

The results are summarized in Table 2. We see that, with a significance level of 0.01, the null-hypotheses are cleanly rejected for both the CITE and MULTI datasets, but this is clearly not the case for their cross CITE×MULTI; this difference can also be seen in Figure 16c. A possible explanation for the consistency within datasets not carrying over across datasets could be the following.

DATASET	N_{TOTAL}	$N_{SUCCESS}$	H_0 PROB.	STATISTIC	P-VALUE
CITE	2,359	1,601	0.627	0.679	$1.1 \cdot 10^{-7}$
MULTI	2,747	2,031	0.668	0.739	$2.4 \cdot 10^{-16}$
CITE×MULTI	1,368	765	0.560	0.559	0.528

Table 2: Results of the binomial tests that test the *consistency* of the two verdicts of a pair $p \in \mathcal{O}$; for both datasets and their cross. The null hypothesis claims the equality of Equation 12, the alternative claims the left-hand side to be greater.

The full data from a single dataset implies a certain structure of causal relationships that the FCI algorithm performed on its sub-clusters is indeed able to find, explaining the internal consistency. However, this implied structure does not accurately reflect the underlying process that generated the data, explaining why we find different causal relationships when we repeat the process based on new data. This idea is reinforced by high degree of noise observed in the data; noise can shape the causal structure implied by the data, which is lost when considering another dataset, but is retained when looking at an overlapping cluster in the same dataset.

This is rather poor news for our effort to map the flow of genetic information in a cell. However, due to the highly significant results for each single dataset in Table 2, we can still safely conclude that performing FCI on small subsets of the full variable-space retains at least some of the causal relationships implied by the full space. This is important, as it justifies to a degree the representation of a global causal network by the sum of its parts; a key piece of our approach to tackling cyclical causality in high-dimensional spaces.

5.2 LIMITATIONS AND FUTURE RESEARCH

The most obviously limitation is the lack of consistent causal graphs across datasets, but, as mentioned in the previous section, that might be an indication of a general difficulty the FCI algorithm has in discerning the true causal DMG from the noisy data. For future research, a suggestion would be to give the algorithm a bit more foothold to help it along. We could, for example, assume that adjacencies in an underlying DMG can only occur between different modalities. This would be justified by considering the primary flow of genetic information: DNA⟶RNA⟶protein⟶DNA. In the context of a single dataset, this would reduce to either RNA⟶protein⟶RNA or DNA⟶RNA⟶DNA for CITE and MULTI respectively; in either case, adjacencies are expected to be between differing modalities. From the outset, it is not entirely clear how one would incorporate this kind of extra knowledge in the FCI algorithm. However, that may be worth investigating, as it would significantly reduce the number of possible adjacencies; especially considering that each dataset is dominated by a single modality (RNA) in terms of variable count. As such, it could limit the influence of noise on the output.

Also the significant results in Table 2 deserve some scrutiny. Though the observed consistency is much better than mere chance, it is far from perfect; 0.679 and 0.739 for CITE and MULTI respectively. If we were planning to use the numerous CDPAGs as building blocks to create a single network spanning the entire variable-space, then it seems likely that we would require perfect or near-perfect consistency between these building blocks. A suggestion for future research would be to handle this as follows. Instead of performing FCI on each cluster independently; perform them one by one, each time moving to an overlapping cluster and taking

any previously identified information as background information to the FCI algorithm. In this way the consistency between clusters is a given, and the use of additional information may also lead to more conclusive FCI output. The obvious drawback of this is that the consistency between clusters can no longer be used to verify the found results. This drawback, instead of ruling out the suggested method, should instead be interpreted as a symptom of a more general limitation of this work; one that can be fixed as well.

It concerns the following. Both issues outlined above are in some way connected to our effort to solve two problems simultaneously. The first one is the development of a causal inference method that can handle cyclical causality in high-dimensional data; compounded by the additional complication that the data is segregated into two datasets that need to be bridged by their overlap in variables. The second problem is the application of this method to a relatively poorly understood and noisy corpus of data. The lack of a ground truth makes it hard to verify this new method, and the lack of a verified method makes it hard to judge its output.

This brings us to the primary suggestion for future research, which is to take a more principled approach by splitting this problem up into the two separate projects that it deserves. The verification of the method can be done by generating experimental data, given a cyclical DMG $\mathcal{G}$ of high dimension. In order to mimic the segregated nature of the multimodal single-cell data; the generated data can be split into two separate datasets, deleting a subset of the variables in one, and deleting another subset in the other. Since we know the causal graph $\mathcal{G}$ underlying the data, we know the ground truth of this problem to be its σ-independence model $\mathbb{M}_\sigma(\mathcal{G})$. With this ground truth in hand, the verification of our method as well as the impact of the updates suggested above, can be straightforwardly evaluated. Additionally, it would give a sturdy foothold to tweak the method and optimize it for fidelity. Note that there is still a lot of room for variation in the construction of the clusters and in the use of background information. Experimentation with these variations has little value now, but would have considerable value if a ground truth was available to evaluate these variations against.

If and when this method is verified, including any updates and modifications, it can be reapplied to the multimodal single-cell data under consideration here. An additional venue of further research would then be to verify the resulting causal relationships against known relationships from biological research. The relationship between DNA and RNA is fairly well understood; given the identifiers of the genes that we have in our data, they could be matched to their reference sequences, using a source like the NCBI 2023 databases. This should provide actionable information on the relationship between various DNA and RNA genes. Similar information may be found for the relationship between RNA and proteins; and together this could provide a partial approximation of the ground truth behind the observed data.

6

CONCLUSION

We set out to map the causal relationships between the numerous DNA genes, RNA genes, and proteins in a cell. To this end we proposed a two-tiered causal inference approach that would be able to handle high-dimensional data generated by a process that contains cyclical causal relationships. Specifically, we look at the multimodal single-cell data provided by OpenProblems 2022, which has two datasets, *(i)* the CITE dataset containing observations of RNA genes and proteins, and the *(ii)* the MULTI dataset containing observations of DNA genes and RNA genes.

Even though this approach was specifically designed to handle high-dimensional data, the first order of business was still to reduce the dimension of the DNA modality. With data on 200k+ different DNA genes, this modality was pushing the limits of feasibility; with the main bottleneck being the computation of the correlation matrix, requiring a quadratic order of calculations. We reduce this modality to a few hundred variables by combining neighboring genes vis-à-vis their physical location on the chromosomes.

Next was the first tier of our approach, we performed a custom clustering procedure to subdivide the variable-space into partially overlapping clusters. Each cluster was to be small enough to make it manageable for a standard causal inference algorithm, and the overlap would provide a way to link the various clusters together.

The second tier of the approach involved the execution of the FCI algorithm on each of the individual variable subsets defined by the clustering. The output of the FCI algorithm can be interpreted as representing a possibly cyclical DMG. This is shown by Mooij and Claassen 2020, which has served as the central source of theory throughout this work.

To evaluate the results of the execution of these two tiers, we focused on the overlapping parts of the clusters. Since each cluster has its own FCI output that implies certain ancestral relationships between its variables; we could check whether the overlapping parts were judged consistently by different clusters. We indeed found this consistency among overlapping CITE variables, and among overlapping MULTI variables, but not for overlap between CITE variables and MULTI variables.

This led us to believe that, while the two-tiered procedure can be used to identify causal relationships in high-dimensional data driven by cyclical causality; it is not suited, in its current form, to discover the underlying process that drives the multimodal single-cell data. We propose various ways to improve the procedure aimed at its current shortcomings; most prominently we propose to verify the two-tiered approach with data for which the generating process is known, before applying it to the complex and noisy data of intracellular dynamics.

BIBLIOGRAPHY

10xGenomics (2020). *References*. URL: https://support.10xgenomics.com/single-cell-gene-expression/software/release-notes/build (visited on 06/05/2023).

Bianconi, E., A. Piovesan, F. Facchin, A. Beraudi, R. Casadei, F. Frabetti, L. Vitale, M. C. Pelleri, S. Tassani, F. Piva, and S. Perez-Amodio (2013). "An estimation of the number of cells in the human body." In: *Annals of Human Biology* 40.6, pp. 463–471.

Cao, J., 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 (2018). "Joint profiling of chromatin accessibility and gene expression in thousands of single cells." In: *Science* 361, pp. 1380–1385.

Chickering, D. M. (2000). "Learning Equivalence Classes Of Bayesian Network Structures." In: *Journal of Machine Learning Research* 2, pp. 445–498.

— (2002). "Optimal Structure Identification With Greedy Search." In: *Journal of Machine Learning Research* 3, pp. 507–554.

Clancy, S. and W. Brown (2008). "Translation: DNA to mRNA to Protein." In: *Nature Education*. https://www.nature.com/scitable/topicpage/translation-dna-to-mrna-to-protein-393/.

Cusanovich, D. A., A. J. Hill, D. Aghamirzaie, R. M. Daza, H. A. Pliner, Berletch J. B., G. N. Filippova, X. Huang, L. Christiansen, W. S. DeWitt, C. Lee, S. G. Regalado, D. F. Read, F. J. Steemers, C. M. Disteche, C. Trapnell, and J. Shendure (2018). "A Single-Cell Atlas of In Vivo Mammalian Chromatin Accessibility." In: *Cell* 174.5, pp. 1309–1324.

Eckart, C. and G. Young (1936). "The approximation of one matrix by another of lower rank." In: *Psychometrika* 1.3, pp. 211–218.

Ensembl (2023). *Gene annotation in Ensembl*. URL: https://www.ensembl.org/info/genome/genebuild/index.html (visited on 06/05/2023).

Fisher, R. A. (1921). "On the probable error of a coefficient of correlation deduced from a small sample." In: *Metron* 1, pp. 1–32.

Forré, P. and J. M. Mooij (2018). "Constraint-based Causal Discovery for Non-Linear Structural Causal Models with Cycles and Latent Confounders." In: *Conference on Uncertainty in Artificial Intelligence*.

Kaggle (2022). *Open Problems - Multimodal Single-Cell Integration*. URL: https://www.kaggle.com/competitions/open-problems-multimodal/overview (visited on 06/05/2023).

Klein, A. M., L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner (2015). "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells." In: *Cell* 161.5, pp. 1187–1201.

Kotliarov, Y., R. Sparks, A. J. Martins, M. P. Mulè, Y. Lu, M. Goswami, L. Kardava, R. Banchereau, V. Pascual, A. Biancotto, J. Chen, P. L. Schwartzberg, N. Bansal, C. C. Liu, F. Cheung, S. Moir, and J. S. Tsang (2020). "Broad immune activation underlies shared set point signatures for vaccine responsiveness in healthy individuals and disease activity in patients with lupus." In: *Nature Medicine* 26.4, pp. 618–629.

Kramer, M. A. (1991). "Nonlinear principal component analysis using autoassociative neural networks." In: *AIChE Journal* 2, pp. 233–243.

Lance, C., M. D. Luecken, D. B. Burkhardt, R. Cannoodt, P. Rautenstrauch, A. Laddach, A. Ubingazhibov, Z. Cao, K. Deng, S. Khan, Q. Liu, N. Russkikh, G. Ryazantsev, U. Ohler, A. O. Pisco, J. Bloom, S. Krishnaswamy, and F. J. Theis (2021). "Multimodal single cell data integration challenge: results and lessons learned." NeurIPS https://www.biorxiv.org/content/10.1101/2022.04.11.487796v1.

Macosko, E. Z., 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 (2015). "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets." In: *Cell* 161.5, pp. 1202–1214.

Meek, C. (1997). "Graphical models: selecting causal and statistical models." In.

Mooij, J. M. and T. Claassen (2020). "Constraint-Based Causal Discovery In The Presence Of Cycles." In: *Conference on Uncertainty in Artificial Intelligence*. Vol. 124. UAI '20.

NCBI (2023). *National Center for Biotechnology Information*. URL: https://www.ncbi.nlm.nih.gov/ (visited on 08/17/2023).

Open-Problems (2022). *Multimodal Single-Cell Integration Across Time, Individuals, and Batches*. URL: https://openproblems.bio/events/2022-08_neurips/ (visited on 06/05/2023).

Pearson, K. (1901). *LIII. On lines and planes of closest fit to systems of points in space*.

Ramsey, J., M. Glymour, R. Sanchez-Romero, and C. Glymour (2017). "A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images." In: *International Journal of Data Science and Analytics* 3.2, pp. 121–129.

Richardson, T. S. and P. Spirtes (2002). "Ancestral graph Markov models." In: *The Annals of Statistics* 30.4, pp. 962–1030.

Sachs, K., O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan (2005). "Causal protein-signaling networks derived from multiparameter single-cell data." In: *Science* 208, pp. 523–529.

Shannon, C. E. (1948). "A mathematical theory of communication." In: *The Bell System Technical Journal* 27.3, pp. 379–423.

Shendure, J. and H. Ji (2008). "Next-generation DNA sequencing." In: *Nature Biotechnology.* 26.10, pp. 1135–1145.

Spirtes, P. and C. Glymour (1991). "An algorithm for fast recovery of sparse causal graphs." In: *Social science computer review* 9.1, pp. 62–72.

Spirtes, P., C. Glymour, and R. Scheines (2001). *Causation, Prediction, and Search*. The MIT Press.

Stoeckius, M., C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert (2017). "Simultaneous epitope and transcriptome measurement in single cells. Nat Methods." In: *Nature Methods* 14.9, pp. 865–868.

Tang, F., 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 (2009). "mRNA-Seq whole-transcriptome analysis of a single cell." In: *Nature Methods* 6.5, pp. 377–382.

Velten, L., S. F. Haas, S. Raffel, S. Blaszkiewicz, S. Islam, B. P. Hennig, C. Hirche, C. Lutz, E. C. Buss, D. Nowak, T. Boch, W. Hofmann, A. D. Ho, W. Huber, A. Trumpp, M. A. G. Essers, and L. M. Steinmetz (2017). "Human haematopoietic stem cell lineage commitment is a continuous process." In: *Nature Cell Biology* 19.4, pp. 271–281.

Verma, T. and J. Pearl (1990). "Equivalence and Synthesis of Causal Models." In: *Conference on Uncertainty in Artificial Intelligence*, pp. 255–270.

Zhang, J. (2008). "On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias." In: *Artificial Intelligence* 172.16, pp. 1873–1896.

COLOPHON

The template for the layout of this document, `classicthesis`, was developed by André Miede, and is available at https://bitbucket.org/amiede/classicthesis/. The frontpage drawing by Frank Ramspott is available at https://fineartamerica.com.