Towards Learning Cell Causal-Embeddings

Paul Bertin
Mila
Université de Montréal
paul.f.bertin@gmail.com

Sreya Francis
Mila
Université de Montréal
sreyafrancis.mec@gmail.com

Joseph Paul Cohen
Mila
Université de Montréal
joecohen@gmail.com

Yoshua Bengio
Mila
Université de Montréal

Abstract

In many cases, the precise modes of action of drugs at the cellular level remain poorly understood, and a better understanding of the effect of a drug on cells is needed both from the scientific and clinical points of view. In this work we try to make a first step in that direction by making hypothesis specific to gene expressions, in order to take into account the complicated structure of gene expression data, and obtain more meaningful and robust representations. We model the distributions of gene expressions in different environments (*e.g* different drug treatments) and try to learn representations that change in a sparse way from one environment to another. We explore how the proposed models capture the generative process in a simple case and evaluate them out of distribution. We experiment on both synthetic and gene expression data. For this work, we have been advised by J. P. Cohen and Y. Bengio. The code is available at `https://github.com/Bertinus/causal_gene_embedding`

1 Introduction

In many cases, the precise modes of action of drugs at the cellular level remain poorly understood. A better understanding of the effect of a drug on cells, besides the purely scientific interest of the question, would be useful in drug discovery pipelines. The tremendous amounts of gene expression data opens the way to machine learning based techniques, giving the opportunity to get precise insight into the mechanisms of the cell and predict high level phenotypes and drug response.

Several challenges appear when it comes to applying machine learning to such data. The cell is a complicated system, which can be in many different states. Thus lots of confounding effects can appear and are difficult to account for, limiting the usefulness of naive statistical approaches. Indeed, identifying markers of a specific phenotype is insufficient, and identifying genes that are causally related to a phenotype as opposed to correlated with the cause of the phenotype, remains an open challenge [Battle and Montgomery, 2014]).

In order to make robust predictions, one has to take into account the complicated structure of gene expression data. We thus need to make relevant hypothesis specific to gene expression, relying on domain knowledge that will allow us to generalize robustly while keeping enough flexibility. As a comparison with computer vision, where one of the main breakthroughs has been Convolutional Neural Networks, we do not have a good inductive bias for gene expression data (an equivalent of *translation equivariance*).

Several gene interaction graphs that summarize decades of biology research have been built and are publicly available, but it seems that they do not provide useful prior knowledge for machine learning

pipelines in the context of gene expression data [Hashir et al., 2019]. In this work, we try to make some biologically relevant hypothesis that we use to design machine learning models. Moreover, we try to take into account the structure of the dataset by considering that the data comes from different environments, corresponding to different growing media, drug treatments and genomes. We hope that this can allow us to build more meaningful representations of gene expressions and better generalization.

TODO: related multi environment setting to temporal data (multiple "envs" are seen along time, by assuming temporal consistency at a fine scale) and the construction of boolean networks of gene interaction networks (`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3086598/`, `https://arxiv.org/pdf/2006.01023.pdf`)

2 Gene expression

In this section we explain in more details the data we work with and provide some background in genetics.

2.1 RNA-seq

RNA-sequencing is a technique that quantifies the amount of RNA in a population of cells. The quantity of mRNA (messenger Ribonucleic acid) is a proxi for the quantity of proteins that the cells produce at a given time, giving information on the state and properties of the tissue from which cells have been sampled. There can be discrepancies: the concentration of mRNA is not always highly correlated with the concentration of the corresponding protein [Liu et al., 2016b]. There are batch effects and several types of preprocessing are used to account for differences in the lengths of the genes and in the total amount of genetic material [Mortazavi et al., 2008; Li and Dewey, 2011].

2.2 Gene regulation

There exist many types of interactions between genes. The basic mechanism of regulation is the following: a regulatory gene codes for a protein (called transcription factor) that will eventually bind to specific sites of the regulatory sequence of the regulated gene, preventing (or allowing) the *RNA polymerase* from performing transcription [Michalak, 2008]. The regulation can also happen at the level of RNA (*microRNAs* bind to the target RNA, preventing translation). The regulatory sequence can be either close to the coding sequence (*cis*) or far away (*trans*).

There are different types of regulatory elements (binding sites in the regulatory sequence) : promoters (turns on transcription), silencers (turns off transcription), enhancers (promoter of the promoter), insulators (prevents regulatory elements from having an effect on neighbouring genes). Note that gene regulation depends on many factors that are *unobserved* like time and environment of the cell.

2.3 A partially observable system

Of course, RNA sequencing does not allow us to observe all the important variables that govern the state of the cell, thus what we are studying is a partially observable system.

Expression Quantitative Trait Loci (eQTLs) have been identified in different tissues [Dimas et al., 2009; Nica et al., 2011; Myers et al., 2007], confirming that some regions of the non-coding genome have an impact on gene expression patterns. Thus, the mechanisms governing gene expression can change from one sample to another, based on mutations in the non coding genome (which is typically not observed in basic RNA-seq). When experiments are performed on a specific cell line, the genome is supposed to be identical for all cells in the cell line.

The folding of the chromatin also has an effect by preventing some parts of the genome from being expressed as they are folded and not accessible to the *RNA polymerase* (the protein that performs transcription). [Singh et al., 2016] predicts gene expression from histone modification information. Efficiently combining the different types of genomic data (DNA sequence, histone modification, DNA methylation, miRNA, chromatin features) with gene expression data remains a challenge, and some works attempt to do so [Beer and Tavazoie, 2004; Chaudhary et al., 2017; Liu et al., 2016a].

3 Related work

3.1 Representation Learning on Gene Expression Data

Representation learning techniques have been applied to gene expression data. Chen et al. [2015] proposed D-GEX, a Multi Layer Perceptron which infers gene expressions from a set of landmark genes. Denoising auto-encoders have also been aplied [Tan et al., 2016, 2017] and Sharifi-Noghabi et al. [2018] have proposed a specific training procedure that uses both labeled and unlabeled data.

Some works try to take advantage of the structure of latent space of machine learning models. Way and Greene [2017, 2018] analysed the merits of VAEs when applied to gene expression data. Latent space arithmetics seem to give interesting insight. Du et al. [2018] applied Word2Vec[Mikolov et al., 2013] to gene expression data and defined the environment of a gene as the set of genes which are highly correlated with it. [Asgari and Mofrad, 2015] used Word2Vec where gene neighbourhoods correspond to biophysical and biochemical properties of the sequence.

Other works try to get a more precise understanding of the structure of the data. Techniques to learn gene embeddings that can predict protein-protein interaction and other pair-wise information have been proposed [Cao et al., 2017]. The authors of [Trofimov et al., 2017] find factorized embeddings that allow to accurately predict a gene expression given a sample embedding and a gene embedding.

3.2 Cell response to perturbation

Variational autoencoders along with latent space vector arithmetics have been used to predict drug response [Dincer et al., 2018; Lotfollahi et al., 2019]. The authors of [Wang et al., 2019] use an autoencoder with hard constraint on the latent representation to learn a representation which is disentangled from the drug effect point of view.

3.3 Multi-environment training and invariance

Previous works have explored multi-environment training procedures There exist tight links between invariance and causality [Bühlmann, 2018] and they have lead to the development of new methodologies to achieve better robustness such as Anchor regression [Rothenhäusler et al., 2018]. Recently, Arjovsky et al. [2019] have proposed to learn representations from which classifiers can be optimal across all environments.

Compared to these settings, we are interested in an unsupervised problem. It prevents us from enforcing complete invariance between environments, which would ignore the useful information that comes from the differences between environments. Some work have tried to distinguish between environment specific and environment invariant features using an adversarial setting to impose invariance [Ebrahimi et al., 2020], but in our context, we argue that any latent feature would be susceptible to change.

TODO: talk about ICP: `https://arxiv.org/pdf/1501.01332.pdf` and see this `https://arxiv.org/pdf/2006.07433.pdf`

Latent tree VAE `https://arxiv.org/pdf/2006.07433.pdf`

Talk by Caroline Uhler: `https://www.youtube.com/watch?v=s8w3mOBtEgk&feature=youtu.be`

4 Methods

We would like to explore new methods that allow to share information between environments by having *lots* of properties preserved between them while being flexible enough to allow any of the features to have a different distribution in a given environment. We will first present a few definitions and then state the hypothesis we make. Finally we will present the models we propose.

4.1 Definitions

Definition 1 *A **Structural Causal Model** or **SCM** [Peters et al., 2017, Chapter 6] consists of a collection of assignments :*

$$C_j \coloneqq f_j(\mathbf{PA}_j, Z_j), \qquad j = 1...n$$

where $\mathbf{PA}_j \subseteq \{C_1, ..., C_n\} \backslash \{C_j\}$ are called the parents of C_j, and of a joint distribution P_Z over the noise variables $Z_1, ..., Z_n$ which are jointly independent. Let us consider the directed graph $\mathcal{G}$ whose vertex set is $\{C_1, ..., C_n\}$ and in which there is an edge from each parent in $\mathbf{PA}_j$ to C_j, for all $j = 1, ..., n$. $\mathcal{G}$ is assumed to be acyclic.

Note that the noise variables $Z_1, ..., Z_n$ completely describe the variability in the observations generated by the SCM.

Definition 2 *A **noise intervention** on a variable j is a soft intervention where we only change the distribution of the noise associated with variable j (no change in f_j).*

Definition 3 *We call **latent SCM** a Structural Causal Model over a set of unobserved variables $C = \{C_1, ..., C_n\}$ such that observed variables $\{X_1, ..., X_{n_g}\}$ are generated from C in an invariant manner, i.e. $P(X) \coloneqq g(C)$ where g is invariant and cannot be modified by any intervention.*

4.2 TODO

Add a word about the identifiability of the network. If we consider that the network is a tree, and there is only one intervened variable at a time, can we identify the structure?

Let us assume the causal variables are known. If the effect of a node on its children is non zero for each child and for all nodes, yes? For each causal variable, we can count the number of environments in which it changes. It gives us a topological order among variables. We take the first one (root), all variables that change are children of the root (i.e. they are in the same tree). Apply this recursively.

What if causal variables are unknown? It should depend on the family of function we use for the decoder (versus the "true" decoder).

4.3 Structure of the datasets

We consider $D \coloneqq \bigcup_{e \in \varepsilon} D_e$ a dataset of gene expression data acquired in different environments $e \in \varepsilon$. The data corresponding to each environment is denoted $D_e \coloneqq \{x_i^e\}_{i=1}^{n_e}$ and assumed to be Independent Identically Distributed. Here $x_i^e \in \mathbb{R}^{n_g}$ is a vector of length the number of genes whose expression has been acquired (n_g can range from 1k to 20k).

4.4 Hypothesis on the dataset

Hypothesis 1: latent SCM There exists a latent Structural Causal Model over a set of d unobserved variables $C = (C_i)_{i=1}^d$ such that the observed gene expressions are a function of C. Note that C is itself a function of the noise variables Z associated with the SCM.

Hypothesis 2: sparsity of change between environments A given environment corresponds to a noise intervention on one of the C_i variables. The prior distribution over Z thus changes in a sparse way from one environment to the other.

The change from one environment to the other is sparse in some representation, a given drug acts on only a few mechanisms, not everything changes at the same time.

Hypothesis 3: environment representations We assume that we have access to a representation e of each environment, that contains information about the way the distribution of the variables C changes.

Hypothesis 4: no selection bias between environments We hypothesize that there is no confounding effect between the environment and gene expressions, that is to say the effect of the environment on the gene expression is blocked by the variables C. Therefore, we have $X \perp\!\!\!\perp e | C$. This hypothesis

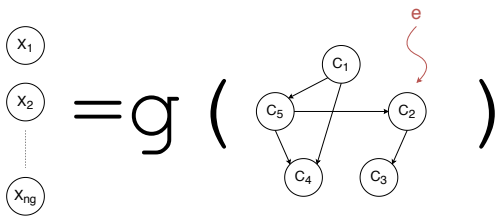


Figure 1: Overview of our assumptions. The environment e has a sparse effect on the representation C. The observations X are a function of C, and this functions does not change across environments.

is relevant for lab experiment data but may be violated for clinical data (drug assignments can be biased).

4.5 Environment Variational Encoder

Let us present the model we propose. Following our hypothesis, there exists a representation C whose prior distribution changes in a sparse way between environments.

We want to model:

$$P_{\theta,\phi}(X|e) = \int P_\theta(X|C,e)P_\phi(C|e)dc \tag{1}$$

Following our hypothesis, we have $X \perp\!\!\!\perp e|C$ so $P_\theta(X|C,e) = P_\theta(X|C)$. The parameters can be learned by maximizing the variational lower bound:

$$\log P_{\theta,\phi}(X|e) \geq \mathbf{E}_{C\sim Q_\psi(C|X,e)}[\log P_\theta(X|C)] - \mathbf{KL}[Q_\psi(C|X,e)||P_\phi(C|e))] \tag{2}$$

where the encoder Q_ψ is a model with parameters ψ that approximates the posterior probability $P(C|X,e)$. For now, we will model the prior distribution over C in a given environment as a diagonal multivariate Gaussian $P_\Phi(C|e) = \mathcal{N}(\mu_e, \sigma_e^2)$, where μ_e and σ_e are environment dependent parameters. We will predict μ_e and σ_e with an auxiliary network which takes a representation of the environment as input, typically a vector embedding of the drug molecule.

In future work, we could try to model interactions between the variables C and predict the variability between samples by inferring the noise variables Z of the SCM. We would have the following factorization $P_\Phi(C|e) = P(C|Z)P(Z|e)$ with $P(C|Z) = \Pi_i P(C_i|Z_i, C_1, ..., C_{i-1})$ and $P_\phi(Z|e)$ a diagonal multivariate Gaussian $P_\phi(Z|e) = \mathcal{N}(\mu_e, \sigma_e^2)$.

We also want the prior distributions to differ in a *sparse* way from one environment to the other. In order to do so, we will enforce the prior $P_\phi(C|e)$ in one environment to be close to the reference Gaussian $\mathcal{N}(0,1)$ in all but one dimension, *i.e.*, we want all components but one of μ_e to be zero (or almost zero), and all components but one of $\log(\sigma_e)$ to be zero (or almost zero). Therefore, we normalize each component of μ_e and σ_e with a *Softmax*. We have the normalized mean $\bar{\mu}_e = \mu_e \otimes \text{Softmax}(|\mu_e|)$ where $\otimes$ is the elementwise product and $\text{Softmax}(x)_i = \frac{e^{(1/T)x_i}}{\sum_j e^{(1/T)x_j}}$ with temperature T. Similarly $\overline{\log(\sigma_e)} = \log(\sigma_e) \otimes \text{Softmax}(|\log(\sigma_e)|)$. The lower the temperature T, the more we enforce sparsity.

TODO: check that we have the absolute value in the code (inside the softmax)

During training, we have to compute the Kullback-Leibler divergence w.r.t $\mathcal{N}(\mu_e, \text{diag}[\sigma_e^2])$ instead of $\mathcal{N}(0,1)$ as in the regular Variational Autoencoder. The precise derivation is presented in Appendix A.

4.6 Advantages of the model

Comparison with the Conditional VAE Compared to the Conditional VAE [Sohn et al., 2015], the decoder does not take the environment representation as input, which will force the representation

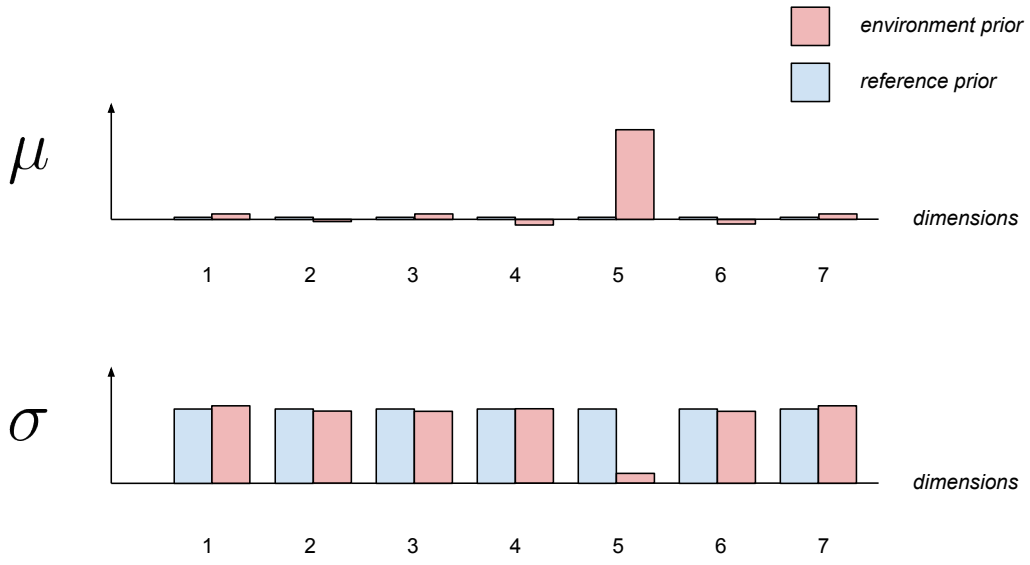


Figure 2: Sparse difference between the environment prior and the reference prior which is the standard Gaussian. Here the difference is concentrated along dimension 5. Edit figure to show (Softmax(|mul)) and give a name to it

C to contain the relevant information about the environment. We further enforce that environments are *close* from one another, pushing the model towards learning latent variables that are disentangled from the environment effect point of view (e has a sparse effect on C).

Drug effect prediction Once provided with an environment, we get a generative model that is adapted to the new environment. If the environment contains information about the molecule, the drug dosage, the cell line, and tissue type, we can get gene expression predictions for an unseen combinations of characteristics. Moreover, if the environment representation e has some structure (*e.g.* molecular fingerprints) we could even hope to be able to accurately predict the effect of unseen drugs.

4.7 Additions to the basic model

To further enforce the decoder to *learn* the right causal variables, we can add an Invariant Risk Minimization regularization [Arjovsky et al., 2019] to the parameters of the decoder so that it becomes an invariant predictor of gene expressions.

Several conditionings are possible for the encorder. Instead of giving the representation e as input to the encoder, we could use other conditioning such as FiLM [Perez et al., 2017].

4.8 Translation Variational Autoencoder

Note that in the previous setting, we only enforce the sparsity of change from one environment to another by some term in the loss function (in the KL term to be more precise). We propose a variant of the Environment Variational Autoencoder where we predict an offset μ_e which is environment specific and added to the latent variables $C_{\text{prev}} \sim \mathcal{N}(0, 1)$ predicted by the encoder. The sum $C = C_{\text{prev}} + \mu_e$ is fed to the decoder.

In this setting, the latent variables C follow a prior distribution $\mathcal{N}(\mu_e, 1)$ in a given environment. The encoder just predicts the factors of variation which are independent of the environment. The sparsity of μ_e is enforced with a Softmax normalization in the same way as before.

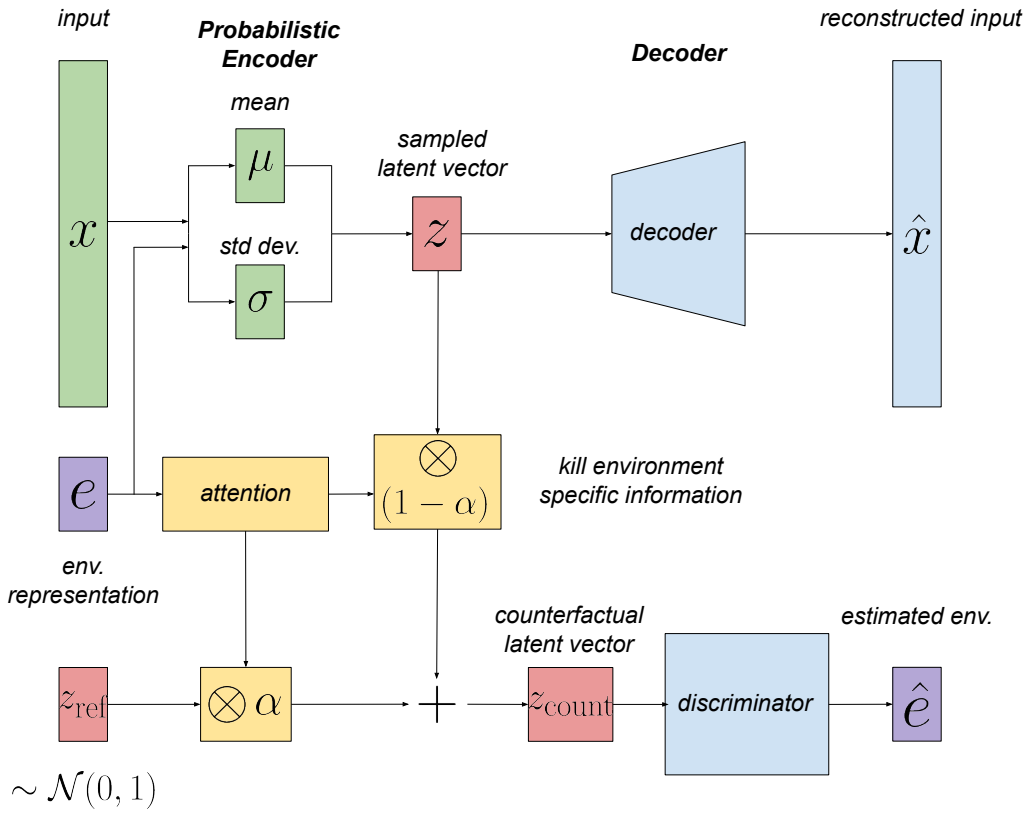


Figure 3: Schematic view of the Adversarial Environment Autoencoder

4.9 Structured Models

The encoder predicts the noise and we use an auto-regressive model to generate the causal variables. We can use model of depth n and width k to simulate any (?) DAG whose tree has depth < n and width < k

4.10 Adversarial Prior Autoencoder

The sparse change of the prior can be learnt adversarially. Let $Z \in \mathbb{R}^k$ be the output of the encoder. Let $\alpha \in \mathbb{R}^k$ be an attention vector predicted by an auxiliary model A_η taking the environment representation e as input (we have $\sum_i \alpha_i = 1$). Let $\varepsilon \sim \mathcal{N}(0, \mathrm{I}_k)$ be some Gaussian noise.

Let $\otimes$ be the element-wise product. We define a *counterfactual* representation $Z_{\text{count}} = \alpha(e) \otimes \varepsilon + (1 - \alpha(e)) \otimes Z$. This counterfactual representation is almost the same as the original Z for most elements, while the elements on which the attention mechanism is focused are *brought back* closer to the reference Gaussian.

The *counterfactual* representation Z_{count} is fed to an adversarial model D_γ that tries to predict the environment representation from Z_{count}.

$$\mathcal{L}_{\text{AP-AE}} = \mathbf{E}_{\substack{Z \sim Q_\psi(Z|X,e) \\ \varepsilon \sim \mathcal{N}(0,\mathrm{I}_k)}} [\log P_\theta(X|Z) - \log D_\gamma(e|\alpha_\eta \otimes \varepsilon + (1 - \alpha_\eta) \otimes Z)]$$

We optimize:

$$\min_{\gamma} \max_{\psi,\theta,\eta} \mathcal{L}_{\text{AP-AE}}$$

TODO: maybe formalize what counterfactual we compute exactly. see `https://ftp.cs.ucla.edu/pub/stat_ser/r485.pdf` We enforce that the environment always has a localized effect.

In words, the adversarial network will try to predict from which environment the *counterfactual* representation came from. The encoder will try to infer embeddings which have sparse differences from one environment to the other, and the attention mechanism will try to focus on the elements that are different from usual in a given environment.

`https://arxiv.org/pdf/1909.04443.pdf`

`https://arxiv.org/pdf/1511.05644.pdf`

TODO: noise injection in GANS: `https://arxiv.org/pdf/2006.05891.pdf` interesting link with the structured models we wanna do

5 Datasets

5.1 Synthetic dataset

A presentation of the synthetic dataset is available online[1]. This synthetic dataset is based on a graph and designed to mimic some properties of gene expressions, such as gene regulation. Latent variables and observed variables depend on their parents $V_i \sim f(\text{PA}_i, \varepsilon)$ with PA_i the parents of V_i. The structural functions f can be chosen to be linear or neural networks. Moreover, one can apply interventions on the latent variables (hard or soft) resulting in downstream effects on the children of the intervened variable, and possibly complicated changes in observation space.

5.2 Lincs L1000

Lincs L1000 is a bulk RNA-seq dataset where we have a access to several perturbagens and cell lines. This dataset[2] is available on the GEO plateform. The entire dataset consists of 1000 gene expression profiles for thousands of perturbagens collected over a variety of doses and cell lines and includes 118050 samples, corresponding to individual experiments with a single compound across 12328 genes, out of which 978 are landmark genes (used to infer the remaining 11350). The environment representation we use is the Morgan fingerprint of the perturbagen.

Metadata (cell line infos):

modification: ['-666', 'genetically modified to stably express Cas9 protein', 'genetically modified to stably express Cas9 protein', 'immortalized normal', 'bone marrow cells that were immortalized', 'hTERT-immortalized normal kidney cells immunoselected for DBA-positivity', 'differentiated from ESC to be motor neurons', 'terminally differentiated to be neurons', 'NEU exposed to KCl (potassium chloride) solution to activate neurons', 'differentiated from iPSC, but not terminally differentiated', 'NPC that were genetically modified to stably express Cas9 protein']

donor age: [54, 58, -666, 74, 31, 15, 36, 44, 50, 56, 69, 51, 61, 52, 47, 55, 73, 33, 62, 43, 60, 29, 81, 1, 37]

donor sex: M, F

donor ethnicity: ['-666', 'Caucasian', 'Black']

original source vendor: ['ATCC', 'DSMZ', 'ECACC', '-666', 'HSRRB', 'JCRB', 'RIKEN', 'KCLB', 'Harvard University', 'Sciencell', 'CellzDirect', 'Lonza', 'LONZA']

original growth pattern : ['adherent', '-666', 'suspension', 'mix']

[1] `https://github.com/Bertinus/causal_gene_embedding/blob/master/Notebooks/datagenerator_presentation.ipynb`

[2] L1000 Connectivity Map perturbational profiles from Broad Institute LINCS Center for Transcriptomics: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138

subtype: [’malignant melanoma’, ’non small cell lung cancer| carcinoma’, "ewing’s sarcoma", ’adenocarcinoma’, ’carcinoma’, ’colorectal adenocarcinoma’, ’non small cell lung cancer| large cell carcinoma’, ’carcinoma| epithelial-mucinous’, ’non small cell lung cancer| adenocarcinoma’, ’mucinous papillary adenocarcinoma’, ’normal kidney’, ’non small cell lung cancer| squamous cell carcinoma’, ’colorectal carcinoma’, ’endometrial adenocarcinoma’, ’embryonal kidney’, ’epithelial’, ’hepatocellular carcinoma’, ’acute myelogenous leukemia (AML)| M3 (promyelocytic)’, ’-666’, ’bone marrow’, ’normal endothelial cell|umbilical cord’, ’endometrial adenocarcinoma| endometrioid carcinoma’, ’acute lymphoblastic leukemia (ALL)| T-cell’, ’carcinoma| prostate’, ’skin fibroblast’, ’small cell lung cancer| carcinoma’, ’non small cell lung cancer| adenosquamous carcinoma’, ’kidney epithelial’, ’acute myeloid leukemia (AML)’, ’acute myelogenous leukemia| promyelocytic’, ’carcinoma| clear cell’, ’neuroblastoma’, ’acute myelogenous leukemia| M5 (monocytic)’, ’carcinoma| undifferentiated’, ’myeloman, haematopoietic,lymphoid’, ’osteosarcoma’, "lymphoma| B-cell| non-hodgkin’s| histiocytic", "lymphoma| B-cell| non-hodgkin’s| diffuse large cell", ’pancreatic carcinoma’, ’normal stem fibroblast-derived iPScs’, ’normal primary adipocyte stem cells’, ’normal primary liver’, ’myoblast’, ’normal primary skeletal muscle cells’]

primary site: [’skin’, ’lung’, ’bone’, ’stomach’, ’breast’, ’large intestine’, ’ovary’, ’kidney’, ’endometrium’, ’liver’, ’haematopoietic and lymphoid tissue’, ’vascular system’, ’prostate’, ’autonomic ganglia’, ’blood’, ’pancreas’, ’-666’, ’central nervous system’, ’adipose’, ’muscle’]

sample type: [’tumor’, ’-666’, ’normal’, ’primary’]

Metadata (gene infos):

gene title: e.g. ribosomal protein, kinase, peptidase, splicing factor

is landmark gene: 1, 0

is bing (best infered gene): 1, 0

Metadata (perturbation infos):

pert ID: e.g. BRD-K08703257

canonical smiles

inchi-key: e.g. DOMWKUIIPQCAJU-JKPPDDDBSA-N

name: e.g. acecainide

pert type: [’trt cp’, ’ctl vehicle’, ’trt xpr’, ’ctl untrt’, ’ctl vector’]

Metadata (signature infos):

quote : Molecular signatures are sets of genes, proteins, genetic variants or other variables that can be used as markers for a particular phenotype. Reliable signature discovery methods could yield valuable insight into cell biology and mechanisms of human disease.

signature ID: contains the plate ID (the multiwell plate) used for the experiment, the cell line and the incubation time

pert ID

pert name

pert type

cell id

pert idose: e.g. ’8.0 um’ (micromolar, a concentration unit) (about 10 percent of unknown value)

pert itime: [’24 h’, ’3 h’, ’6 h’, ’96 h’]

distil ID: seems to summarize all previous infos in one string

Metadata (signature metrics):

sig ID

distil cc q75: a float between 0 and 1

distil ss: a float between 4 and 13

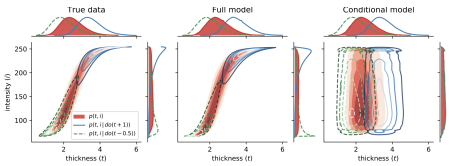


Figure 3: Distributions of thickness and intensity in the true data (left), and learned by the full (centre) and conditional (right) models. Contours depict the observational (red, shaded) and interventional joint densities for $\mathrm{do}(t := f_T(\epsilon_T) + 1)$ (blue, solid) and $\mathrm{do}(t := f_T(\epsilon_T) - 0.5)$ (green, dashed).

Figure 4: TODO: for synthetic dataset draw things like this `https://arxiv.org/pdf/2006.06485.pdf`

pert ID

pert iname

pert type

tas: a float between 0.1 and 0.55

ngenes modulated up lm: integer

ngenes modulated down lm: integer

pct self rank q25: a float (positive)

distil nsample: intger between 1 and 6

6 Experiments

We want to investigate whether models learn latent variables which are disentangled from the environment effect point of view. We call latent variables disentangled from the environment effect point of view if the marginals $P(C_i|e)$ are almost all the same from one environment to the other. We first investigate that point on synthetic experiments with 2 latent variables. We then present general results on more complex synthetic data and on the L1000 dataset.

6.1 Linear case with 2 independent latent variables

We first try some synthetic experiments in a very simple setting. Let us have 2 latent variables and 10 observed variables which are functions of the latent. Each observed variable has a probability 0.5 to have a given latent variable as a parent. If an observed variable has no parent, we add a parent at random among latent variables. All structural functions are linear with coefficients chosen in $\{-1, 1\}$.

We can apply a translation to each of the latent variables to change the distribution of the observed one. We represent a given environment as (T_0, T_1), where T_i is the offset added to latent variable i when generating the data.

We first train a regular Variational Autoencoder (VAE) in one environment $(0, 0)$. We use a linear decoder and an encoder with one hidden layer of 5 neurons with ReLU activations. At test time, we can visualize how well a model generalizes out of distribution by computing the average reconstruction loss in environments $(T_0, 0)$ (direction 0) and $(0, T_1)$ (direction 1). We vary T_0 and T_1 between -20 and 20 and report the results in Figure 5.

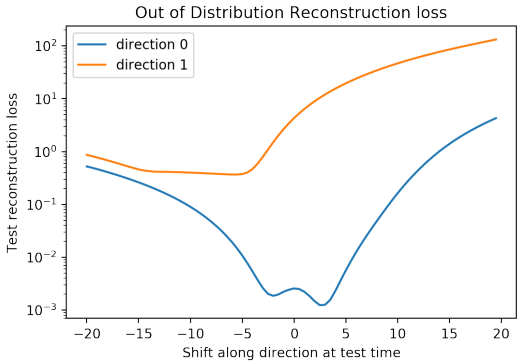
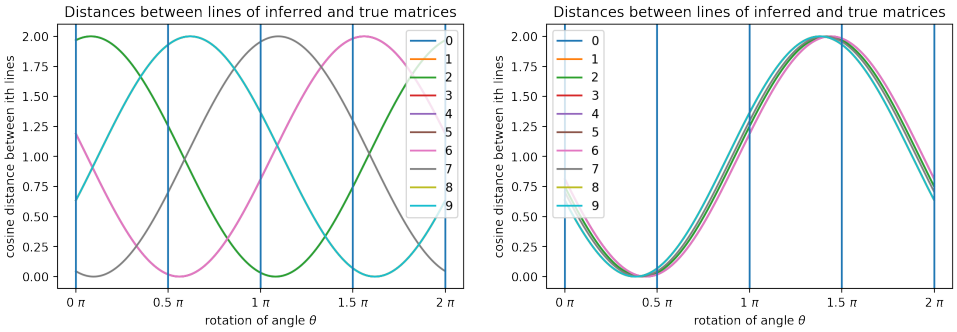


Figure 5: Basic VAE. Out of distribution reconstruction loss in environments $(T_0, 0)$ (direction 0) and $(0, T_1)$ (direction 1) for T_0 and T_1 varying between -20 and 20. Case where latent variables are independent and training on one environment $(0, 0)$.

We want to investigate in more details how well the VAE infers the true causal variables. In this setting, the observations are generated by $X = M.C$ with C the true latent variables and $M \in \mathbb{R}^{10\times 2}$ the true weight matrix. At inference time we identify $X = \tilde{M}.\tilde{C}$. Note that any couple $\tilde{M} = M.R_\theta.\text{diag}(\lambda)$, $\tilde{C} = \text{diag}(1/\lambda).R_{-\theta}.C$ is solution. Here R_θ is the rotation matrix of $\mathbb{R}^2$ with angle θ and $\lambda \in \mathbb{R}^2$. Let us check whether M and $\tilde{M}$ match for some θ and λ to confirm that we can identify true latent variables up to a scaling and rotation factor.

To do so, we rotate the matrix M by an angle of θ and obtain $M.R_\theta$. We then compute the cosine distance between the i^{th} line of $M.R_\theta$ and the i^{th} line of $\tilde{M}$ for all $\theta \in [0, 2\pi]$ and all $i \in [0, 9]$. The results are reported in Figure 6(a). In order to also consider the cases where the elements of λ have different signs, we perform the same experiment after applying a symmetry to the first column of M (we multiply all the elements of the first column by -1). The results are reported in Figure 6(b)



(a) Cosine Distances without symmetry

(b) Cosine Distances with symmetry

Figure 6: Basic VAE. Cosine distance between the i^{th} line of $M.R_\theta$ and the i^{th} line of $\tilde{M}$ for all $\theta \in [0, 2\pi]$ and all $i \in [0, 9]$. In Figure 6(b), the optimal angle is the same for all lines and is around 0.4π. Case where latent variables are independent and training on one environment $(0, 0)$.

We see that when a symmetry is applied, the optimal angle is the same for all lines and is around 0.4π, meaning that M and $\tilde{M}$ match for a rotation angle of 0.4π and a scaling factor whose components have opposite signs.

Note that the inferred latent variables are disentangled from the environment effect point of view if and only if the optimal angle between M and $\tilde{M}$ is a multiple of $\pi/2$. That is to say, a translation along one of true latent variables should correspond to a change in only one of the inferred latent variables. This is not the case in Figure 6 as expected, because we trained on only one environment and we use the basic VAE for which there is no pressure towards learning disentangled embeddings.

6.2 Linear case with 2 dependent latent variables

We now perform the same experiment in the degenerate case where the second latent variable is a deterministic function of the first one. The out of distribution reconstruction losses are presented in Appendix Figure 11 and the inferred and ground truth matrices $\tilde{M}$ and M are compared in Appendix Figure 12.

Now there are many degrees of freedom, and an observed variable that is in fact a function of the first latent variable can be equally well predicted from the second latent variable. In practice, we see that $\tilde{M}$ and M do not match for any angle, which is coherent with the point we just made.

We also see in Figure 11 that the model achieves bad out of distribution generalization in some directions.

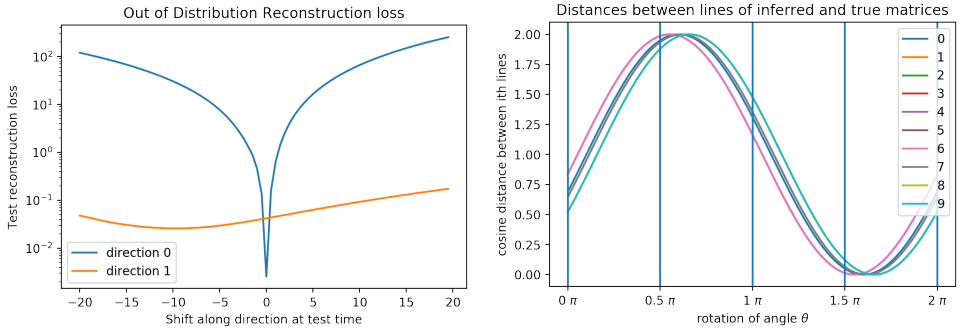
6.3 Linear case with 2 dependent latent variables and 2 environments

We know perform the same experiment again with two dependent latent variables but training on two different environments, $(0, 0)$ and $(0, 0.5)$ where an offset of 0.5 is applied to the second variable in the second environment.

We present similar experiments as before in Appendix Figures 13 and 14. Training on two environments for which the offset of the second latent variable is different adds more constraint, and M and $\tilde{M}$ match again for some θ (although the match is less clean than in Figure 6). Moreover, we can see that the model achieves good out of distribution generalization in Appendix Figure 13.

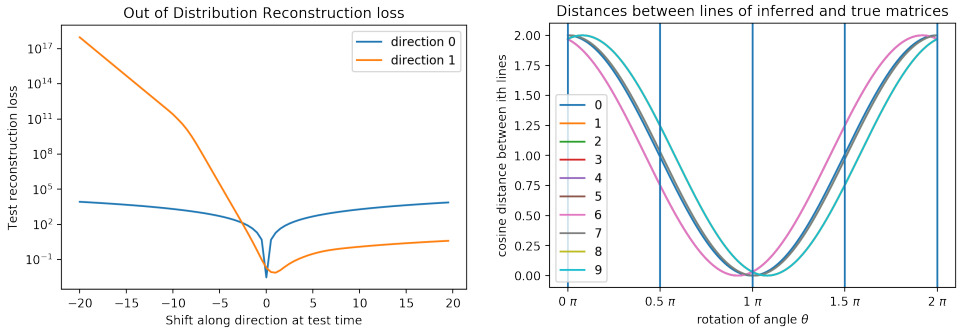
We performed similar experiments with Conditional VAE and Environment VAE. The results are reported in Figure 7 and Figure 8. Compared to the basic VAE, those models tend to overfit and reconstruct poorly out of distribution, even in regions close to the training distribution.

We should definitely investigate further those behaviours in simple settings such as this one.



(a) Out of distribution reconstruction (b) Cosine Distances with symmetry

Figure 7: Environment VAE. Degenerate case where the second latent variable is a deterministic function of the first one and training on two environments $(0, 0)$ and $(0, 0.5)$.



(a) Out of distribution reconstruction

(b) Cosine Distances with symmetry

Figure 8: Conditional VAE. Degenerate case where the second latent variable is a deterministic function of the first one and training on two environments $(0, 0)$ and $(0, 0.5)$.

6.4 Non linear case with 5 dependent latent variables and 5 random environments

We perform experiments with our synthetic data generator, but this time we use 5 latent variables linked together randomly by a *growing network* directed graph. The structural equations between latent variables are noisy linear: latent variables are linear functions of their parents plus some gaussian noise. The structural functions of the observed variables are 1 hidden layer neural networks with 3 hidden units and ReLU activations. The weights of all structural functions are chosen at random in $\{-1, 1\}$.

The model are trained on 5 environments chosen at random. To generate an environment, a latent variable is chosen at random and an offset sampled from $\mathcal{N}(0, 5)$ is applied, having downstream effects on the children. The validation set comes from 5 other random environments. The validation reconstruction losses along training are reported in Figure 9.

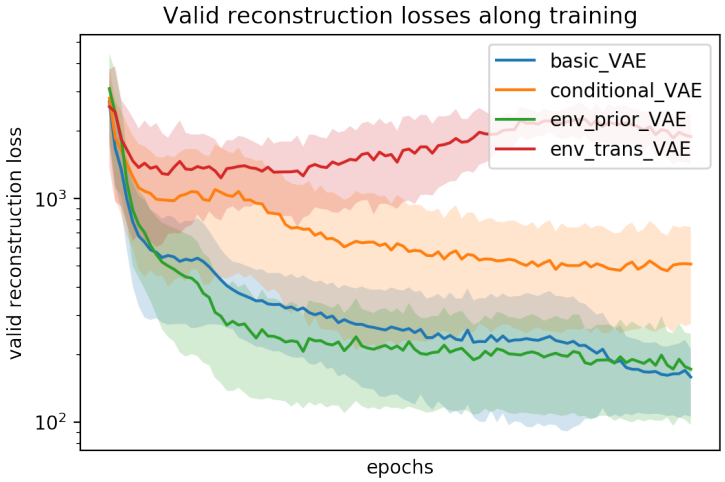


Figure 9: Out of distribution validation reconstruction loss along training for non linear synthetic data. The basic VAE and Environment VAE (env prior) perform well compared to other models, while the conditional VAE and the Translation VAE (env trans) do not perform as well. Standard deviations computed over 3 trials.

6.5 Experiments on the L1000 dataset

Finally, we perform experiments on the L1000 dataset. We train on 682 environments (14631 samples) and validate on 335 environments (7167 samples). All validation environments are unseen during

training. The validation reconstruction losses along training are reported in Figure 10. The input is 978 dimensional, and the latent space is 200 dimensional for all models. All encoders, decoders and auxiliary networks for models which have one are one hidden layer neural networks with 500 hidden units and ReLU activations.

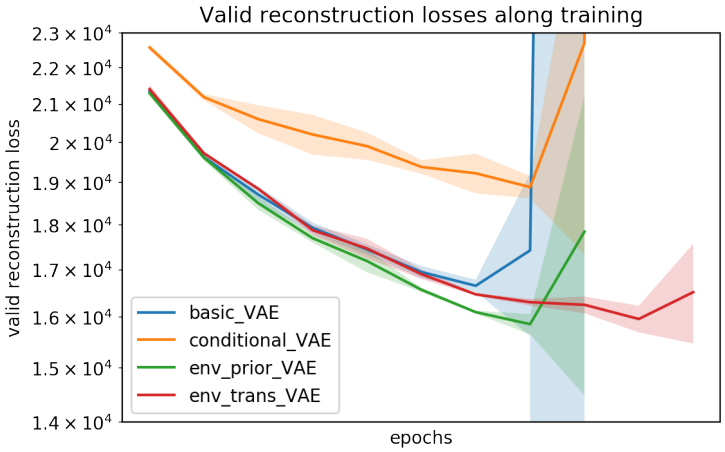


Figure 10: Out of distribution validation reconstruction loss along training on the L1000 dataset. All losses tend to explode after some time. Standard deviations computed over 3 trials.

6.6 Validation

learn the graph in a similar way to what has been done in ADAGE + - compare to DRKG - plug it in the pipeline we have for gene-interaction graph analysis

7 Conclusion

We tried to investigate how taking into account multiple environments in the model design while trying to push the model towards learning sparse differences between environments could allow us to build better representations. We should build upon our current results in order to have a clearer view of the performance of each model, and figure out why losses explode on the L1000 dataset.

More work is needed to understand how to disentangle variables from the drug effect point of view, and how to obtain sparse models if the real generative process is sparse. We need to evaluate our models in a more precise manner, either by investigating in details the structure of the latent space and relate it to biological knowledge, or by evaluating our models on downstream tasks (although the transfer from one gene expression dataset to another is difficult due to different preprocessings).

We also want to model interactions between latent variables as mentioned in section 4.5. Other directions to investigate would be to explore multi-environment training procedures [Bengio et al., 2019; Arjovsky et al., 2019] and self-supervised learning, although the lack of straight-forward way to do data-augmentation in gene expression data makes the use of self-supervised learning difficult.

Integreate in an active pipeline? `https://arxiv.org/pdf/2006.05690.pdf`

TODO: Drug database `https://www.drugbank.ca/`

Gene set analysis `http://www.webgestalt.org/`

Have a look at Adage: We could try to infer gene-gene interactions from the model, and relate it to known databases as STRING.

TODO: data normalization??

KEGG `https://www.genome.jp/ftp/dbget/README`

`https://github.com/gnn4dr/DRKG`

We could identify gene gene interactions in a similar way as in ADAGE. Moreover, we could have some dynamic interactions if the decoder is not linear (for a given sample, what is d(gene j)/d(hidden unit i)?)

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *CoRR*, abs/1907.02893.

Asgari, E. and Mofrad, M. R. K. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287.

Battle, A. and Montgomery, S. (2014). Determining causality and consequence of expression quantitative trait loci. *Human genetics*, 133.

Beer, M. A. and Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell*, 117(2):185–198.

Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. J. (2019). A meta-transfer objective for learning to disentangle causal mechanisms. *CoRR*, abs/1901.10912.

Bühlmann, P. (2018). Invariance, causality and robustness.

Cao, J., Wu, Z., Ye, W., and Wang, H. (2017). Learning functional embedding of genes governed by pair-wised labels. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, page 397–401. IEEE.

Chaudhary, K., Poirion, O. B., Lu, L., and Garmire, L. X. (2017). *Deep Learning based multi-omics integration robustly predicts survival in liver cancer*.

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2015). *Gene expression inference with deep learning*.

Dimas, A. S., Deutsch, S., Stranger, B. E., Montgomery, S. B., Borel, C., Attar-Cohen, H., Ingle, C., Beazley, C., Arcelus, M. G., Sekowska, M., and et al. (2009). Common regulatory variation impacts gene expression in a cell type dependent manner. *Science (New York, N.Y.)*, 325(5945):1246–1250.

Dincer, A. B., Celik, S., Hiranuma, N., and Lee, S.-I. (2018). *DeepProfile: Deep learning of cancer molecular profiles for precision medicine*.

Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., and Zhi, D. (2018). Gene2vec: Distributed representation of genes based on co-expression. *bioRxiv*.

Ebrahimi, S., Meier, F., Calandra, R., Darrell, T., and Rohrbach, M. (2020). Adversarial continual learning.

Hashir, M., Bertin, P., Weiss, M., Frappier, V., Perkins, T. J., Boucher, G., and Cohen, J. P. (2019). Is graph-based feature selection of genes better than random? *arXiv:1910.09600 [cs, q-bio]*. arXiv: 1910.09600.

Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics*, 12:323.

Liu, F., Li, H., Ren, C., Bo, X., and Shu, W. (2016a). Pedla: predicting enhancers with a deep learning-based algorithmic framework. *Scientific Reports*, 6(1):28517.

Liu, Y., Beyer, A., and Aebersold, R. (2016b). On the dependency of cellular protein levels on mrna abundance. *Cell*, 165(3):535–550.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2019). scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721.

Michalak, P. (2008). Coexpression, coregulation, and cofunctionality of neighboring genes in eukaryotic genomes. *Genomics*, 91(3):243–248.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628.

Myers, A. J., Gibbs, J. R., Webster, J. A., Rohrer, K., Zhao, A., Marlowe, L., Kaleem, M., Leung, D., Bryden, L., Nath, P., and et al. (2007). A survey of genetic human cortical gene expression. *Nature Genetics*, 39(12):1494–1499.

Nica, A. C., Parts, L., Glass, D., Nisbet, J., Barrett, A., Sekowska, M., Travers, M., Potter, S., Grundberg, E., Small, K., and et al. (2011). The architecture of gene regulatory variation across multiple human tissues: The muther study. *PLoS Genetics*, 7(2).

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. (2017). Film: Visual reasoning with a general conditioning layer. *arXiv:1709.07871 [cs, stat]*. arXiv: 1709.07871.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. (2018). Anchor regression: heterogeneous data meets causality.

Sharifi-Noghabi, H., Liu, Y., Erho, N., Shrestha, R., Alshalalfa, M., Davicioni, E., Collins, C. C., and Ester, M. (2018). *Deep Genomic Signature for early metastasis prediction in prostate cancer*.

Singh, R., Lanchantin, J., Robins, G., and Qi, Y. (2016). Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648.

Sohn, K., Lee, H., and Yan, X. (2015). *Learning Structured Output Representation using Deep Conditional Generative Models*, page 3483–3491. Curran Associates, Inc.

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

Tan, J., Hammond, J. H., Hogan, D. A., and Greene, C. S. (2016). Adage-based integration of publicly available pseudomonas aeruginosa gene expression data with denoising autoencoders illuminates microbe-host interactions. *mSystems*, 1(1):e00025–15, /msys/1/1/e00025–15.atom.

Trofimov, A., Cohen, J. P., Perreault, C., Bengio, Y., and Lemieux, S. (2017). Uncovering the gene usage of human tissue cells with joint factorized embeddings. In *Workshop on Computational Biology ICML*.

Wang, Z., Yeo, G. H. T., Sherwood, R., and Gifford, D. (2019). *Disentangled Representations of Cellular Identity*, volume 11467, page 256–271. Springer International Publishing.

Way, G. P. and Greene, C. S. (2017). Evaluating deep variational autoencoders trained on pan-cancer gene expression. *arXiv:1711.04828 [q-bio]*. arXiv: 1711.04828.

Way, G. P. and Greene, C. S. (2018). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 23:80–91.

A Kullback-Leibler divergence derivation

Derivation of the Kullback-Leibler divergence with $p = \mathcal{N}(\mu(x), \Sigma(x))$ and $q = \mathcal{N}(\mu_e, \Sigma_e)$. Note that when $t \sim \mathcal{N}(m, \Sigma)$ we have $\mathbb{E}(t^T A t) = \mathbf{Tr}(A\Sigma) + m^T A m$.

$$
\begin{aligned}
&\mathbf{KL}(\mathcal{N}(\mu(x), \Sigma(x))||\mathcal{N}(\mu_e, \Sigma_e)) \\
&= \int \mathcal{N}(t, \mu(x), \Sigma(x)) \log\left(\frac{\mathcal{N}(t, \mu(x), \Sigma(x))}{\mathcal{N}(t, \mu_e, \Sigma_e)}\right) \mathrm{d}t \\
&= \frac{1}{2}\int \left[(t-\mu_e)\Sigma_e^{-1}(t-\mu_e) - (t-\mu(x))\Sigma(x)^{-1}(t-\mu(x)) + \log\left(\frac{\det \Sigma_e}{\det \Sigma(x)}\right)\right] p(t)\mathrm{d}t \\
&= \frac{1}{2}\log\left(\frac{\det \Sigma_e}{\det \Sigma(x)}\right) + \frac{1}{2}\mathbf{Tr}(\mathbb{E}_p\left[(t-\mu_e)\Sigma_e^{-1}(t-\mu_e)\right]) \\
&\quad - \frac{1}{2}\mathbf{Tr}(\mathbb{E}_p\left[(t-\mu(x))\Sigma(x)^{-1}(t-\mu(x))\right]) \\
&= \frac{1}{2}\log\left(\frac{\det \Sigma_e}{\det \Sigma(x)}\right) - \frac{1}{2}\mathbf{Tr}(\mathbb{I}_d) + \frac{1}{2}(\mu(x)-\mu_e)\Sigma_e^{-1}(\mu(x)-\mu_e) + \frac{1}{2}\mathbf{Tr}(\Sigma_e^{-1}\Sigma(x)) \\
&= \frac{1}{2}\left[\log\left(\frac{\det \Sigma_e}{\det \Sigma(x)}\right) - d + (\mu(x)-\mu_e)\Sigma_e^{-1}(\mu(x)-\mu_e) + \mathbf{Tr}(\Sigma_e^{-1}\Sigma(x))\right]
\end{aligned}
$$

As our covariance matrices are diagonales, we have

$$
\begin{aligned}
&\mathbf{KL}(\mathcal{N}(\mu(x), \mathrm{diag}[\sigma^2(x)])||\mathcal{N}(\mu_e, \mathrm{diag}[\sigma_e^2])) \\
&= \frac{1}{2}\left[2\sum_{i=1}^{d}\log(\sigma_{e,i}) - 2\sum_{i=1}^{d}\log(\sigma_i(x)) - d + \sum_{i=1}^{d}\frac{1}{\sigma_{e,i}^2}(\mu_i(x)-\mu_{e,i})^2 + \sum_{i=1}^{d}\frac{\sigma_i(x)^2}{\sigma_{e,i}^2}\right]
\end{aligned}
$$

B Linear case with 2 dependent latent variables and 1 training environment

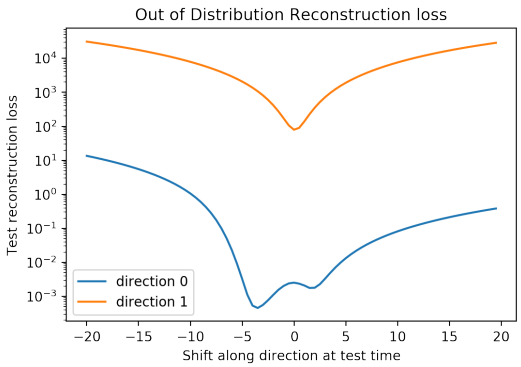
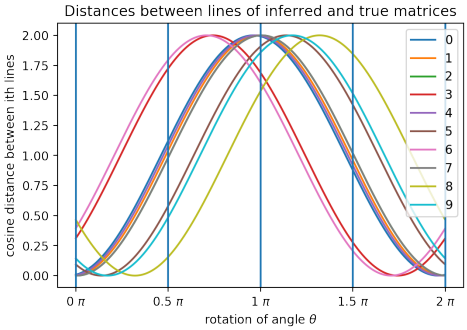
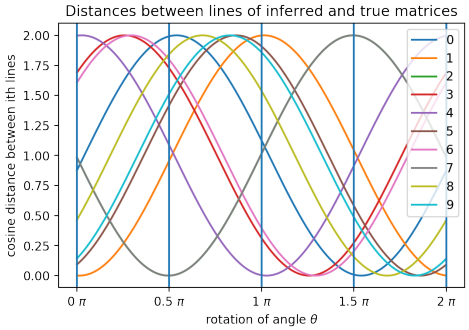


Figure 11: Basic VAE. Out of distribution reconstruction loss in environments $(T_0, 0)$ (direction 0) and $(0, T_1)$ (direction 1) for T_0 and T_1 varying between -20 and 20. Degenerate case where the second latent variable is a deterministic function of the first one and training in one environment $(0, 0)$.



(a) Cosine Distances without symmetry



(b) Cosine Distances with symmetry

Figure 12: Basic VAE. Cosine distance between the i^{th} line of $M.R_\theta$ and the i^{th} line of $\tilde{M}$ for all $\theta \in [0, 2\pi]$ and all $i \in [0, 9]$. In this case, M and $\tilde{M}$ do not match for any angle θ and scaling factor λ. Degenerate case where the second latent variable is a deterministic function of the first one and training in one environment $(0, 0)$.

C Linear case with 2 dependent latent variables and 2 training environments

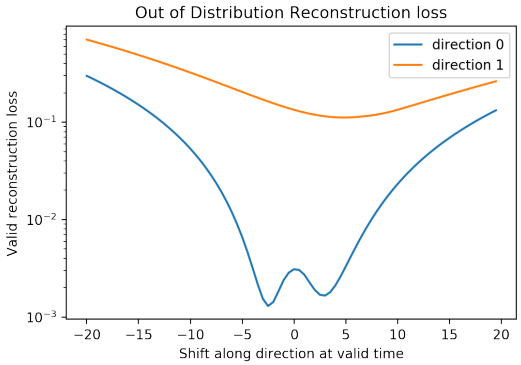
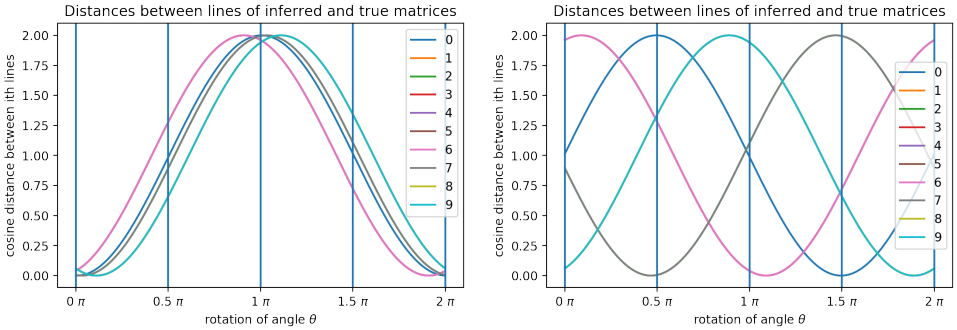


Figure 13: Basic VAE. Out of distribution reconstruction loss in environments $(T_0, 0)$ (direction 0) and $(0, T_1)$ (direction 1) for T_0 and T_1 varying between -20 and 20. Degenerate case where the second latent variable is a deterministic function of the first one and training on two environments $(0, 0)$ and $(0, 0.5)$.



(a) Cosine Distances without symmetry

(b) Cosine Distances with symmetry

Figure 14: Basic VAE. Cosine distance between the i^{th} line of $M.R_\theta$ and the i^{th} line of $\tilde{M}$ for all $\theta \in [0, 2\pi]$ and all $i \in [0, 9]$. In this case, M and $\tilde{M}$ do not match for any angle θ and scaling factor λ. Degenerate case where the second latent variable is a deterministic function of the first one and training on two environments $(0, 0)$ and $(0, 0.5)$.