

Gene expression

Adversarial generation of gene expression data

Ramon Viñas^{1,2,*}, Helena Andrés-Terré¹, Pietro Liò¹ and Kevin Bryson^{2,*} 

¹Department of Computer Science and Technology, University of Cambridge, Cambridge, UK and ²Department of Computer Science, University College London, London, UK

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on January 29, 2020; revised on January 12, 2021; editorial decision on January 14, 2021; accepted on January 20, 2021

Abstract

Motivation: High-throughput gene expression can be used to address a wide range of fundamental biological problems, but datasets of an appropriate size are often unavailable. Moreover, existing transcriptomics simulators have been criticized because they fail to emulate key properties of gene expression data. In this article, we develop a method based on a conditional generative adversarial network to generate realistic transcriptomics data for *Escherichia coli* and humans. We assess the performance of our approach across several tissues and cancer-types.

Results: We show that our model preserves several gene expression properties significantly better than widely used simulators, such as SynTReN or GeneNetWeaver. The synthetic data preserve tissue- and cancer-specific properties of transcriptomics data. Moreover, it exhibits real gene clusters and ontologies both at local and global scales, suggesting that the model learns to approximate the gene expression manifold in a biologically meaningful way.

Availability and implementation: Code is available at: <https://github.com/rvinas/adversarial-gene-expression>.

Contact: rv340@cam.ac.uk or k.bryson@ucl.ac.uk

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Over the last two and a half decades, the emergence of technologies, such as spotted microarrays (Schena *et al.*, 1995), Affymetrix microarrays (Irizarry *et al.*, 2003) and RNA-seq (Mortazavi *et al.*, 2008), has enabled the expression level of thousands of genes from a biological sample to be simultaneously measured. The resulting high-throughput gene expression data can be used to uncover disease mechanisms (Cookson *et al.*, 2009; Emilsson *et al.*, 2008; Gamazon *et al.*, 2018), propose novel drug targets (Evans and Relling, 2004; Sirota *et al.*, 2011), provide a basis for comparative genomics (Colbran *et al.*, 2019) and address a wide range of fundamental biological problems.

However, collecting experimental gene expression data is expensive and datasets of an appropriate size are often unavailable. In these cases, synthetically generated data are often used to benchmark gene expression analysis algorithms. A particular important example of this is evaluating algorithms that reverse engineer gene regulatory networks (GRNs) from transcriptomics data (Irrthum *et al.*, 2010; Margolin *et al.*, 2006; Yu *et al.*, 2004). Benchmarking the performance of these methods is challenging because we often lack well-understood biological networks to use as gold standards. As a result, the current approach is to generate synthetic transcriptomics datasets from well-characterized networks (Schaffter *et al.*, 2011; Van den Bulcke *et al.*, 2006). However, current simulators have been criticized because they fail to emulate key properties of gene expression data (Maier *et al.*, 2013), suggesting that GRN reconstruction

algorithms that perform well on synthetic datasets might not necessarily generalize well on real data.

In this article, we study the problem of generating *in-silico*, realistic transcriptomics data. This is a challenging task, since biological systems are highly complex and it is not clear how biological elements interact with each other. Moreover, it is difficult to determine to what extent the expression data generated by a simulator is realistic. Unlike in other domains, such as image generation, wherein one can empirically assess whether an image is realistic, we do not have an intuitive understanding of high-dimensional expression data.

To address this challenge, we develop a model based on a Wasserstein generative adversarial network with gradient penalty (WGAN-GP) (Gulrajani *et al.*, 2017). In contrast to existing gene expression simulators, such as SynTReN (Van den Bulcke *et al.*, 2006) or GeneNetWeaver (GNW) (Schaffter *et al.*, 2011), our model learns to approximate the expression manifold in a data-driven way and does not require the underlying GRN as input. Furthermore, our approach integrates sample covariates, such as age, sex and tissue-type (global determinants of gene expression) (Stegle *et al.*, 2012) to account for their non-linear effects.

As a first case study, we investigate to what extent the proposed framework preserves statistical properties of GRNs. To that end, we develop a transcriptomics simulator for the *Escherichia coli* bacterium, which has the largest amount of experimentally validated regulatory interactions of any organism (Gama-Castro *et al.*, 2016). We show that our model conserves several gene expression properties significantly better than widely used simulators, such as SynTReN

or GNW. In particular, we introduce several correlation-based metrics to assess the quality of the synthetic data and find that SynTReN and GNW poorly preserve correlations between transcription factors (TFs) and target genes (TGs). This is undesirable and has important implications on the assessment of the ability of GRN reconstruction algorithms to generalize to real data.

As a second case study, we examine whether our approach can be used to generate realistic human gene expression data. Concretely, we train our model on human RNA-seq data from the Genotype-Tissue Expression (GTEx) and The Cancer Genome Atlas (TCGA) and produce data that preserves the tissue and cancer-specific properties of transcriptomics data. Moreover, we observe that the synthetic data conserves gene clusters and ontologies both at local and global scales, suggesting that the model learns to approximate the gene expression manifold in a biologically meaningful way. Finally, we propose a tool that leverages the *in-silico* simulator to find *candidate* causal biomarkers for a variety of cancer-types.

2 Materials and methods

In this section, we introduce our approach to generating realistic gene expression data. Throughout the remainder of the article, we use script letters to denote sets (e.g. \mathcal{D}), upper-case bold symbols to denote matrices or random variables (e.g. \mathbf{X}) and lower-case bold symbols to denote column vectors (e.g. \mathbf{x} or $\bar{\mathbf{q}}_j$). The rest of the symbols (e.g. \bar{q}_{jk} , G or f) denote scalar values or functions.

2.1 Problem formulation

Consider a dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{r}, \mathbf{q})\}$ of samples from an unknown distribution $\mathbb{P}_{\mathbf{x}, \mathbf{r}, \mathbf{q}}$, where $\mathbf{x} \in \mathbb{R}^n$ represents a vector of gene expression values; n is the number of genes; and $\mathbf{r} \in \mathbb{R}^k$ and $\mathbf{q} \in \mathbb{N}^c$ are vectors of k quantitative (e.g. age) and c categorical covariates (e.g. tissue-type or gender), respectively. Our goal is to produce realistic gene expression samples by modelling the conditional probability distribution $\mathbb{P}(\mathbf{X} = \mathbf{x} | \mathbf{R} = \mathbf{r}, \mathbf{Q} = \mathbf{q})$. By modelling this distribution, we can sample data for different conditions and quantify the uncertainty of the generated expression values.

2.2 Adversarial model

Our method builds on a WGAN-GP (Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017). Similar to generative adversarial networks (GAN) (Goodfellow *et al.*, 2014), WGAN-GPs estimate a generative model via an adversarial process driven by the competition between two players, the *generator* and the *critic*.

2.2.1 Generator

The generator aims at producing samples from the conditional $\mathbb{P}(\mathbf{X} | \mathbf{R}, \mathbf{Q})$. Formally, we define the generator as a function $G_\theta : \mathbb{R}^n \times \mathbb{R}^k \times \mathbb{N}^c \rightarrow \mathbb{R}^n$ parametrized by θ that generates gene expression values $\hat{\mathbf{x}}$ as follows:

$$\hat{\mathbf{x}} = G_\theta(\mathbf{z}, \mathbf{r}, \mathbf{q}), \quad (1)$$

where $\mathbf{z} \in \mathbb{R}^u$ is a vector sampled from a fixed noise distribution \mathbb{P}_z and u is a user-definable hyperparameter.

2.2.2 Critic

The critic takes gene expression samples $\bar{\mathbf{x}}$ from two input streams (the generator and the data distribution) and attempts to distinguish the true input source. Formally, the critic is a function $D_\omega : \mathbb{R}^n \times \mathbb{R}^k \times \mathbb{N}^c \rightarrow \mathbb{R}$ parametrized by ω that we define as follows:

$$\bar{y} = D_\omega(\bar{\mathbf{x}}, \mathbf{r}, \mathbf{q}),$$

where the output \bar{y} is an unbounded scalar that quantifies the degree of realism of an input sample $\bar{\mathbf{x}}$ given the covariates \mathbf{r} and \mathbf{q} (e.g. high values correspond to real samples and low values correspond to fake samples).

2.2.3 Optimization

We optimize the generator and the critic adversarially. Following Arjovsky *et al.* (2017), we train the generator G_θ and the critic D_ω to solve the following minimax game based on the Wasserstein distance:

$$\begin{aligned} \min_{\theta} \max_{\omega} \mathbb{E}_{\mathbf{x}, \mathbf{r}, \mathbf{q} \sim \mathbb{P}_{\mathbf{x}, \mathbf{r}, \mathbf{q}}} [D_\omega(\mathbf{x}, \mathbf{r}, \mathbf{q})] - \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z} [D_\omega(\hat{\mathbf{x}}, \mathbf{r}, \mathbf{q})] \\ \text{subject to} \quad \|D_\omega(\mathbf{x}_i, \mathbf{r}, \mathbf{q}) - D_\omega(\mathbf{x}_j, \mathbf{r}, \mathbf{q})\| \leq \|\mathbf{x}_i - \mathbf{x}_j\|, \quad (2) \\ \forall \mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n, \mathbf{r} \in \mathbb{R}^k, \mathbf{q} \in \mathbb{N}^c \end{aligned}$$

where $\hat{\mathbf{x}}$ is defined as in Equation (1) and the constraint enforces the critic D_ω to be 1-Lipschitz, i.e. the norm of the critic's gradient with respect to \mathbf{x} must be at most one everywhere.

Let $\{(\mathbf{x}_i, \mathbf{r}_i, \mathbf{q}_i)\}_{i=1}^k$ be a mini-batch of k independent samples from the training dataset \mathcal{D} . Let $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ be a set of k vectors sampled independently from the noise distribution \mathbb{P}_z and let us define the synthetic samples corresponding to the mini-batch as $\hat{\mathbf{x}}_i = G_\theta(\mathbf{z}_i, \mathbf{r}_i, \mathbf{q}_i)$ for each i in $[1, 2, \dots, k]$. We solve the minimax problem described in Equation (2) by interleaving mini-batch gradient updates for the generator and the critic, optimizing the following problems:

$$\begin{aligned} \text{Generator :} \quad \min_{\theta} \quad & -\frac{1}{k} \sum_{i=1}^k D_\omega(\hat{\mathbf{x}}_i, \mathbf{r}_i, \mathbf{q}_i) \\ \text{Critic :} \quad \min_{\omega} \quad & \frac{1}{k} \sum_{i=1}^k D_\omega(\hat{\mathbf{x}}_i, \mathbf{r}_i, \mathbf{q}_i) - D_\omega(\mathbf{x}_i, \mathbf{r}_i, \mathbf{q}_i) \quad (3) \\ & + \frac{\lambda}{k} \sum_{i=1}^k (\|\nabla_{\hat{\mathbf{x}}_i} D_\omega(\hat{\mathbf{x}}_i, \mathbf{r}_i, \mathbf{q}_i)\|_2 - 1)^2, \end{aligned}$$

where λ is a user-definable hyperparameter and each $\hat{\mathbf{x}}_i$ is a random point along the straight line that connects \mathbf{x}_i and $\hat{\mathbf{x}}_i$, i.e. $\hat{\mathbf{x}}_i = \alpha_i \mathbf{x}_i + (1 - \alpha_i) \hat{\mathbf{x}}_i$ with $\alpha_i \sim \mathcal{U}(0, 1)$. Intuitively, since enforcing the 1-Lipschitz constraint everywhere [see Equation (2)] is intractable (Virmaux and Scaman, 2018), the second term of the critic problem is a relaxed version of the constraint that penalizes the gradient norm along points in the straight lines that connect real and synthetic samples (Gulrajani *et al.*, 2017).

2.2.4 Architecture

Figure 1 shows the architecture of both players. The generator G receives a noise vector \mathbf{z} as input (green box) as well as sample covariates \mathbf{r} and \mathbf{q} (orange boxes) and produces a vector $\hat{\mathbf{x}}$ of synthetic expression values (red box). The critic D takes either a real gene expression sample \mathbf{x} (blue box) or a synthetic sample $\hat{\mathbf{x}}$ (red box), in addition to sample covariates \mathbf{r} and \mathbf{q} , and attempts to distinguish whether the input sample is real or fake. For both players, we use word embeddings (Mikolov *et al.*, 2013) to model the sample covariates (light green boxes), a distinctive feature that allows to learn distributed, dense representations for the different tissue-types and, more generally, for all the categorical covariates $\mathbf{q} \in \mathbb{N}^c$.

Formally, let q_j be a categorical covariate (e.g. tissue-type) with vocabulary size v_j , i.e. $q_j \in \{1, 2, \dots, v_j\}$, where each value in the vocabulary $\{1, 2, \dots, v_j\}$ represents a different category (e.g. lung or kidney). Let $\bar{\mathbf{q}}_j \in \{0, 1\}^{v_j}$ be a one-hot vector such that $\bar{q}_{jk} = 1$ if $q_j = k$ and $\bar{q}_{jk} = 0$ otherwise. Let d_j be the dimensionality of the embeddings for covariate j . We obtain a vector of embeddings $\mathbf{e}_j \in \mathbb{R}^{d_j}$ as follows:

$$\mathbf{e}_j = \mathbf{W}_j \bar{\mathbf{q}}_j,$$

where each $\mathbf{W}_j \in \mathbb{R}^{d_j \times v_j}$ is a matrix of learnable weights. Essentially, this operation describes a lookup search in a dictionary with v_j entries, where each entry contains a learnable d_j -dimensional vector of embeddings that characterizes each of the possible values that q_j can take. To obtain a global collection of embeddings \mathbf{e} , we concatenate all the vectors \mathbf{e}_j for each categorical covariate j :

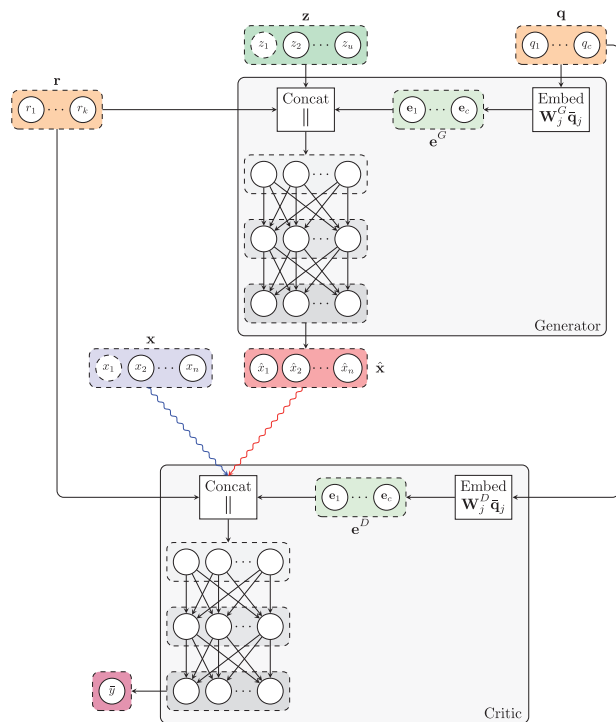


Fig. 1. Architecture of our model. The generator receives a noise vector z , and categorical (e.g. tissue-type; q) and numerical (e.g. age; r) covariates, and outputs a vector of synthetic expression values (\tilde{x}). The critic receives gene expression values from two input streams (real, blue; and synthetic, red) along with the numerical r and categorical q covariates, and produces an unbounded scalar \bar{y} that quantifies the degree of realism of the input samples from the two input streams. A characteristic feature of our architecture is the use of word embeddings e^G and e^D (green boxes) to learn distributed representations of the categorical covariates for both the generator and the critic

$$e = \parallel_{j=1}^c e_j$$

where c is the number of categorical covariates and \parallel represents the concatenation operator. We then use the learnable embeddings e in downstream tasks.

In terms of the player's architecture, we model both the generator G and critic D as neural networks that leverage independent instances e^G and e^D of the categorical embeddings for their corresponding downstream tasks. Specifically, we model the two players as follows:

$$G_\theta(z, r, q) = \text{MLP}(z \parallel r \parallel e^G) \quad D_\omega(\bar{x}, r, q) = \text{MLP}(\bar{x} \parallel r \parallel e^D),$$

where MLP denotes a multilayer perceptron (MLP).

3 Related work

The methodology presented in this article is closely related to Marouf et al. (2020) in that both methods use a WGAN-GP (Arjovsky et al., 2017; Gulrajani et al., 2017) to generate realistic gene expression data. However, our work differs from theirs in several ways. First, while Marouf et al. (2020) focus on generating gene expression data for cells, our method can be used at a higher scale to produce tissue- and organ-specific transcriptomics data. Our approach also works for two different modalities: bulk RNA-seq data and microarray data. Second, the conditioning technique is different in that Marouf et al. (2020) either include an auxiliary classifier or compute the inner product of the class labels and the output features at the critic's output. Instead, we concatenate the sample covariates with the input features and modify the WGAN-GP objective [(Equation (2))] to sample the class labels from the real distribution. We also use word embeddings (Mikolov et al., 2013) to learn

distributed representations for the categories that we condition upon. Finally, the experiments and evaluation metrics (Sections 4 and 5) are substantially different. Specifically, we compare our method with SynTReN and GNW, analyse the clustering and correlation properties of the synthetic data, perform Gene Ontology enrichment analysis and propose a tool to discover *candidate* biomarkers for several cancer-types.

4 Experimental details

Here, we provide an overview of the experimental details. First, we introduce the two datasets on which we evaluate our method: an *E.coli* microarray dataset from the Many Microbe Microarrays Database (M^{3D}) database (Faith et al., 2008) and a dataset of human RNA-seq that integrates data from the GTEx (Aguet et al., 2019) and TCGA (Weinstein et al., 2013). Second, we describe the experimental details, including details about the hyperparameters and training of our model. Finally, we introduce several quantitative metrics that we employ to evaluate whether statistical properties of gene expression are preserved in the generated data.

4.1 Materials

4.1.1 *E.coli* microarray data

To analyse to what extent our model is able to preserve statistical properties of gene regulatory interactions, we introduce a first case study that leverages *E.coli* transcriptomics data from the M^{3D} database. We chose this bacterium because it has a relatively simple genome (~ 4400 genes) and its gene expression mechanisms are well understood (Salgado et al., 2006) and characterized by the RegulonDB database (Gama-Castro et al., 2016). In particular, we selected a meaningful subset of *E.coli* genes whose expression is directly or indirectly regulated by the master regulator cAMP receptor protein (CRP).

4.1.1.1 Many microbe microarrays database. We downloaded *E.coli* single-channel Affymetrix microarray data from the M^{3D} (Faith et al., 2008). From the 7459 available probes, we excluded those corresponding to intergenic regions and controls, resulting in a dataset of 907 samples and 4297 features. These probes were uniformly normalized by Faith et al. (2008) using log-scale robust multi-array average (Irizarry et al., 2003) to reduce batch effects and make the samples comparable across conditions. To scale the data, we applied the standard score, so that the expression values of each gene have mean 0 and standard deviation 1 across all samples.

4.1.1.2 RegulonDB. The GRN of *E.coli* is one of the most well-characterized transcriptional networks of a single cell. RegulonDB (Gama-Castro et al., 2016) is a database that integrates biological knowledge about the transcriptional regulatory mechanisms of *E.coli*. The database gathers information from multiple biological studies to reconstruct the structure of the *E.coli* GRN. We leveraged information from RegulonDB to select the CRP subnetwork of genes and to evaluate the quality of the generated data.

4.1.1.3 CRP hierarchy. To reduce the dimensionality of the dataset and enable learning from a scarce number of samples, we performed breadth-first search on the RegulonDB interactions to select a meaningful subset of genes whose expression is directly or indirectly regulated by CRP. We broke loops by removing non-tree edges as we built the hierarchy. The CRP, which regulates global patterns of transcription in response to carbon availability, is one of the best characterized global transcriptional regulators in *E.coli*.

4.1.2 Human RNA-seq data

We introduce a second case study to analyse the ability of the proposed method to generate human RNA-seq data from a broad range of cancer and normal tissue-types. Specifically, we combined data from GTEx and TCGA, two reference resources for the scientific community that have generated a comprehensive collection of

human transcriptome data in a diverse set of tissues and cancer-types.

4.1.2.1 The GTEx dataset. The GTEx dataset collected transcriptomics data of multiple tissues from around 960 human donors (Aguet *et al.*, 2019). The biospecimen repository includes model systems, such as whole blood and Epstein Barr virus transformed lymphocytes; central nervous system tissues from 13 brain regions; and a wide diversity of other primary tissues from *non-diseased* individuals.

4.1.2.2 The Cancer Genome Atlas. TCGA is a public database that aims to increase the understanding of the genetic basis of a wide range of cancers. The biospecimen repository includes high-throughput genomic data from *diseased* and matched *healthy* samples spanning 33 cancer-types (Weinstein *et al.*, 2013).

4.1.2.3 Data integration. In this study, we specifically selected samples from 15 common tissues in GTEx and TCGA, namely lung, breast, kidney, thyroid, colon, stomach, prostate, salivary, liver, oesophagus muscularis, oesophagus mucosa, oesophagus gastrointestinal, bladder, uterus and cervix. To unify the data and correct for batch effects, we followed the pipeline described by Wang *et al.* (2018). After integrating the data, our dataset consists of 9147 samples and 18 154 genes.

4.2 Implementation details

For the GTEx + TCGA dataset, we included the donor's age as numerical covariate in \mathbf{r} and the tissue-type, sex and condition (cancer or normal) as categorical covariates in \mathbf{q} . For the *E.coli* dataset, we included as covariates the levels of glucose, ampicillin and oxygen; and the temperature, the aeration, the pH and the growth phase of the cell culture. We normalized the numerical variables via the standard score. For each categorical variable $q_j \in \{1, 2, \dots, v_j\}$, we use the rule of thumb $d_j = \lfloor \sqrt{v_j} + 1 \rfloor$ to set all the dimensions of the categorical embeddings for both players.

In terms of the MLP architectures, for the GTEx+TCGA dataset, we used two hidden layers with 256 units for both players. For the *E.coli* dataset, we used two hidden layers with 128 units for both players. For both datasets, we used ReLU activations for the hidden activations and linear output activations for both the generator and critic. The linear activation ensures that the range of the output expression is unrestricted. Adding more hidden layers in the generator or critic networks did not yield significant improvements in our validation scores.

We trained our models using RMSProp (Tieleman and Hinton, 2012) with a learning rate of 0.0005. Regarding the hyperparameter λ to penalize the gradient norm [see Equation (3)], setting $\lambda = 10$, the value recommended by Gulrajani *et al.* (2017), yielded good results. We used early stopping to train the models, stopping when the validation score had not improved in the last 30 epochs. We trained the models for ~ 2 h (GTEx–TCGA) and ~ 10 min (*E.coli*) on a NVIDIA TITAN XP GPU with 12 GB of RAM.

4.3 Evaluating artificial gene expression data

Assessing to what extent simulators are able to generate realistic datasets is a challenging task since we often lack reliable gold standards. Furthermore, unlike for other domains, such as image generation, wherein one can empirically assess whether an image is realistic, we do not have an intuitive understanding of high-dimensional transcriptomics data. In order to evaluate the quality of the synthetic data, in this section, we propose various quality assessment measures that summarize several statistical properties of gene expression.

We first define a similarity coefficient based on the Pearson's correlation coefficient, which we later use to implement the proposed metrics. Let \mathbf{A} be a $n \times n$ symmetric matrix holding the pairwise distances between all genes. In order to measure how faithfully this matrix preserves the pairwise distances with respect to another $n \times n$

distance matrix \mathbf{B} , we define the Pearson's correlation coefficient between the elements in the upper-diagonal of \mathbf{A} and \mathbf{B} :

$$\gamma(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\frac{A_{ij} - \mu(\mathbf{A})}{\sigma(\mathbf{A})} \right) \left(\frac{B_{ij} - \mu(\mathbf{B})}{\sigma(\mathbf{B})} \right)}{\sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^n 1}},$$

where, for a given $n \times n$ matrix \mathbf{G} , $\mu(\mathbf{G})$ and $\sigma(\mathbf{G})$ are defined as:

$$\mu(\mathbf{G}) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n G_{ij}$$

$$\sigma(\mathbf{G}) = \sqrt{\frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n (G_{ij} - \mu(\mathbf{G}))^2}.$$

4.3.1 General metrics

Here, we define generic metrics that can be used for any dataset.

4.3.1.1 Distance between real and artificial distance matrices (S_{dist}). Let $\mathbf{X} \in \mathbb{R}^{m_1 \times n}$ and $\mathbf{Z} \in \mathbb{R}^{m_2 \times n}$ be two matrices containing m_1 real and m_2 synthetic observations for n genes, respectively. For a given distance function d , we define two $n \times n$ distance matrices $\mathbf{D}^{\mathbf{X}}$ and $\mathbf{D}^{\mathbf{Z}}$ as:

$$D_{ij}^{\mathbf{X}} = d(\text{col}(\mathbf{X}, i), \text{col}(\mathbf{X}, j)) \quad D_{ij}^{\mathbf{Z}} = d(\text{col}(\mathbf{Z}, i), \text{col}(\mathbf{Z}, j)), \quad (4)$$

where $\text{col}(\mathbf{X}, i)$ is the i -th column of matrix \mathbf{X} . Throughout the remainder of the article, we use the Pearson's dissimilarity coefficient as the distance function d .

The coefficient $S_{\text{dist}} = \gamma(\mathbf{D}^{\mathbf{X}}, \mathbf{D}^{\mathbf{Z}})$ measures whether the pairwise distances between genes from the real data are correlated with those from the synthetic data.

4.3.1.2 Distance between real and artificial dendrograms (S_{dend}). Let $C: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ be a function that performs agglomerative hierarchical clustering according to a given linkage function, taking a $n \times n$ distance matrix as input and returning the $n \times n$ distance matrix of the resulting dendrogram. Intuitively, each element (i, j) in the dendrogrammatic distance matrices measures the distance between the two outermost clusters that separate genes i and j .

The coefficient $S_{\text{dend}} = \gamma(C(\mathbf{D}^{\mathbf{X}}), C(\mathbf{D}^{\mathbf{Z}}))$ measures the structural similarity between the dendrograms, giving a score close to one when the *real* and *artificial* dendrograms have a similar structure. Consequently, this metric encourages the synthetic distribution to preserve the relationships among groups of genes that are found in the real distribution. Importantly, this coefficient does not necessarily correlate with $\gamma(\mathbf{D}^{\mathbf{X}}, \mathbf{D}^{\mathbf{Z}})$ (see Supplementary Appendix SA for an example).

4.3.2 GRN-specific metrics

The following metrics make use of an a priori known GRN to evaluate statistical properties of gene regulatory interactions.

4.3.2.1 Weighted sum of TF–TG similarity coefficients (S_{TFTG}). Let \mathcal{G} be a function returning the set of indices of the TGs that are regulated by a given TF. For a given dataset \mathbf{D} and a TF f , let $\mathbf{r}_f^{\mathbf{D}}$ be a vector of distances between the expressions of f and the expressions of its TGs:

$$\mathbf{r}_f^{\mathbf{D}} = (d(\text{col}(\mathbf{D}, f), \text{col}(\mathbf{D}, g)) : g \in \mathcal{G}(f))^{\top},$$

where d is an arbitrary distance measure. If the synthetic dataset \mathbf{Z} is realistic with respect to the real dataset \mathbf{X} , the vectors $\mathbf{r}_f^{\mathbf{X}}$ and $\mathbf{r}_f^{\mathbf{Z}}$ will be similar for each TF f in a set of TFs \mathcal{F} . Let w_f be a coefficient associated with the importance of TF f (e.g. we choose $w_f = |\mathcal{G}(f)|$ in the remainder of the article). We summarize this information as follows:

$$S_{\text{TF-TG}}(\mathbf{X}, \mathbf{Z}) = \frac{1}{\sum_{f \in \mathcal{F}} w_f} \sum_{f \in \mathcal{F}} w_f \cdot v(\mathbf{r}_f^{\mathbf{X}}, \mathbf{r}_f^{\mathbf{Z}}),$$

where $v(\mathbf{r}_f^{\mathbf{X}}, \mathbf{r}_f^{\mathbf{Z}})$ is the cosine similarity between vectors $\mathbf{r}_f^{\mathbf{X}}$ and $\mathbf{r}_f^{\mathbf{Z}}$. The coefficient $S_{\text{TF-TG}}(\mathbf{X}, \mathbf{Z})$ measures whether the TF–TG dependencies in the synthetic data resemble those from the real data.

4.3.2.2 Weighted sum of TG–TG similarity coefficients ($S_{\text{TG-TG}}$). Similarly, we define a coefficient $S_{\text{TG-TG}}$ to measure whether the expression of TGs regulated by the same TF in synthetic data conforms well with the analogue expressions in real data:

$$S_{\text{TG-TG}}(\mathbf{X}, \mathbf{Z}) = \frac{1}{\sum_{f \in \mathcal{F}} w_f} \sum_{f \in \mathcal{F}} w_f \sum_{g \in \mathcal{G}(f)} v(\mathbf{q}_{f,g}^{\mathbf{X}}, \mathbf{q}_{f,g}^{\mathbf{Z}}),$$

where, for a given matrix \mathbf{G} , $\mathbf{q}_{f,g}^{\mathbf{G}}$ is the vector of distances between gene g and all the genes regulated by f (excluding g):

$$\mathbf{q}_{f,g}^{\mathbf{G}} = (d(\text{col}(\mathbf{G}, g), \text{col}(\mathbf{G}, i)) : i \in (\mathcal{G}(f) - \{g\}))^{\top}.$$

5 Results

Here, we assess the quality of the synthetic data. First, we evaluate the GAN on the *E.coli* dataset and compare our method to existing approaches for generating *E.coli* expression data from GRNs. Then, we demonstrate the ability of our approach to produce realistic, tissue-specific gene expression for several cancers from GTEx + TCGA.

5.1 *E.coli* evaluation

5.1.1 Baselines

We compared our approaches with other existing methods: SynTReN (Van den Bulcke et al., 2006) and GNW (Schaffter et al., 2011). Given a GRN, these two methods model gene regulatory interactions with ordinary and stochastic differential equations based on Michaelis–Menten and Hill kinetics. These two models have been widely used to produce synthetic gene expression data from GRNs with the purpose of benchmarking network inference algorithms, but have been previously criticized because they fail to emulate key properties of gene expression (Maier et al., 2013). For example, it was shown that clustering genes according to gene expression yields clusters that are significantly different to those of real data, or that the correlations between TFs and TGs are poorly preserved (Maier et al., 2013).

We generated a gene expression dataset of 680 samples both for the GAN, SynTReN and GNW. For SynTReN and GNW, we created a network with 1076 nodes (without background nodes; e.g. external nodes that regulate the expression of genes in the network) corresponding to the CRP hierarchy (see Section 4.1.1). In both cases, we selected the configuration that optimizes the S_{dist} score. For SynTReN, this corresponded to a biological noise of 0.8 out of 1; and experimental noise of 0 (see Supplementary Appendix SB). For GNW, the best coefficient for the noise term of the stochastic differential equations was 0.1 (see Supplementary Appendix SC).

Table 1. Quantitative assessment of the generated data with the results for a *random* and a *real* (M^{3D} train) simulators. Our model achieves the best realism scores (bold values).

Simulator	S_{dist}	S_{dend}	$S_{\text{TF-TG}}$	$S_{\text{TG-TG}}$
<i>Random</i>	0.0000	−0.0002	0.2299	−0.0132
<i>Real</i>	0.9109	0.5197	0.9143	0.9467
SynTReN	0.0449	0.0444	0.2134	0.2594
GNW	0.0587	0.0223	0.1838	0.1930
GAN	0.8145	0.3872	0.8386	0.8734

5.1.2 Statistical properties of regulatory interactions

Table 1 shows a quantitative comparison of the three methods. A lower bound is determined from randomly generated gene expression data following a uniform distribution $\mathcal{U}(0, 1)$. An upper bound is generated from the real data samples from the *E.coli* train dataset. We observe that the proposed model closely approximates the upper bound in every metric, outperforming SynTReN and GNW by a large margin. In fact, SynTReN and GNW perform similar to the random simulator. We attribute this mainly to the fact that SynTReN and GNW rely exclusively on the source GRNs to produce synthetic data. In contrast, the GAN leverages real expression data to build a generative model in an unsupervised manner and does not require any information on the regulatory interactions. In Supplementary Appendix SD, we further analyse differences between the three simulators in terms of the distributions proposed by Maier et al. (2013).

5.2 GTEx + TCGA evaluation

We trained our GAN on the GTEx + TCGA dataset and sampled a synthetic dataset that matches the test set both in number of samples (2287) and proportions of tissue- and cancer-types.

5.2.1 Correlation and cluster analysis

Figure 2 shows the pairwise correlations and dendrograms for 14 important cancer driver genes with high mutation frequency, as described in Bailey et al. (2018). We note that, for this subset of genes, our model closely matches the correlation and clustering expression patterns. To evaluate the clustering quality on a larger scale, we applied k -means to both the test and the generated expression datasets. Figure 3 shows that there exists a bijective mapping between real and synthetic clusters that preserves most of the genes from the real clusters. In other words, for each real cluster, there exists a synthetic cluster that shares the majority of genes (and vice-versa). We further performed an overrepresentation analysis with GOfuncR (Grote, 2020). We note that similar Gene Ontology terms are enriched for each matching pair of gene clusters. Using the real test set as the reference dataset, we computed the metrics from Section 4.3.1. We quantified S_{dist} at 0.920 out of 0.947 and S_{dend} at 0.215 out of 0.222, where the bounds are approximate and given by the metrics applied to the train set. These results suggest that the generated data retains local and global co-expression patterns.

5.2.2 Tissue and cancer-specific gene expression traits

Next, we tested whether the synthetic data accounts for tissue- and cancer-specific traits of gene expression. In particular, we generated a gene expression dataset that matches the statistics of the train set (e.g. size and proportions of tissue- and cancer-types) and used the synthetic data to train a MLP (2 hidden layers of 64 units with ReLU activations) to perform tissue- and cancer-type classification. For tissue-type classification (15 tissues), the scores for the MLP trained on the synthetic data were $AUC = 0.9884 \pm 0.0010$ and $F1 = 0.9222 \pm 0.0040$ (real test set; averaged over 5 runs). The same figures for the MLP trained on real data were $AUC = 0.9986 \pm 0.0003$ and $F1 = 0.9860 \pm 0.0007$. For cancer-normal binary classification, the scores were $AUC = 0.9992 \pm 0.0001$ and $F1 = 0.9893 \pm 0.0009$ for the MLP trained on synthetic data, and $AUC = 0.9997 \pm 0.0001$ and $F1 = 0.9939 \pm 0.0005$ for the MLP trained on real data. Then, we analysed the expression manifold using UMAP (McInnes et al., 2020). Figure 4 shows a UMAP representation of gene expression data across a variety of normal and cancer tissues, combining samples from the test set (unseen during training) with synthetic data produced by the GAN. Overall, these results show that our method is able to emulate tissue- and disease-specific traits of gene expression.

5.2.3 Candidate *causal* biomarkers of cancer-types

Our model affords the opportunity to produce gene expression data for synthetic patients across different tissues and cancer-types. The gene expression data of each patient is fully determined by a latent

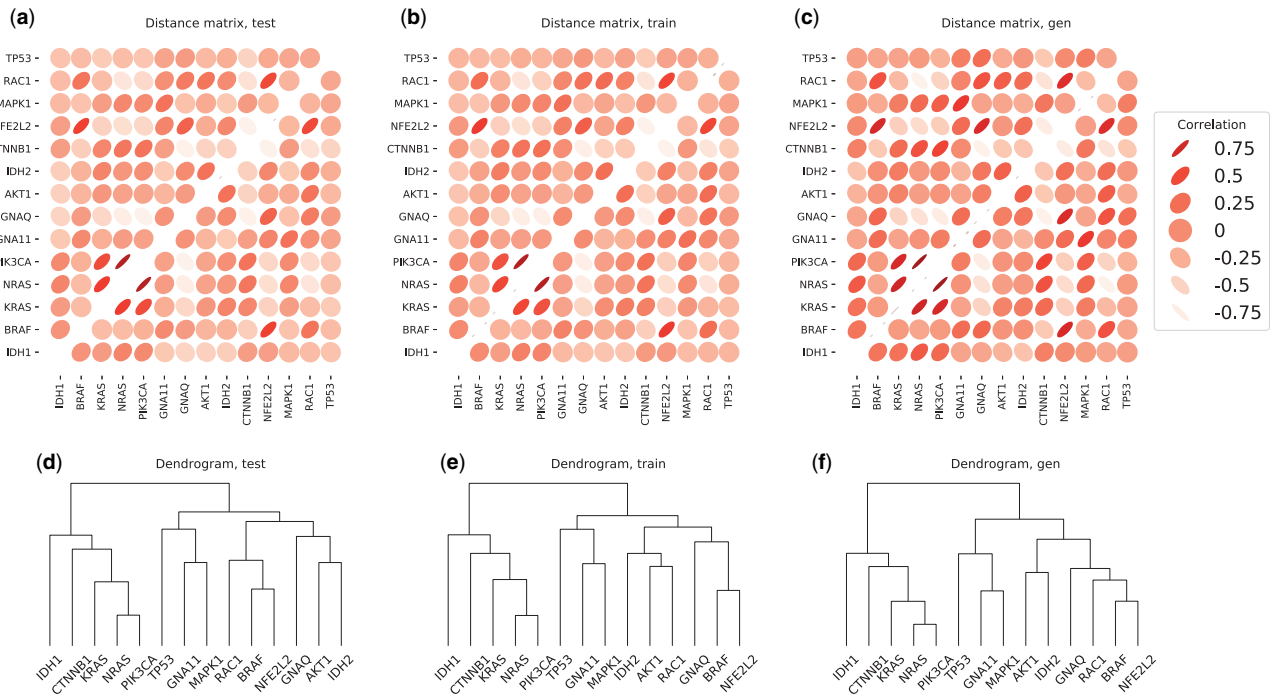


Fig. 2. Correlation matrices and dendrograms for a subset of 14 cancer driver genes with high mutation frequency, as reported in Bailey *et al.* (2018). We use data from the GTEx+TCGA dataset. (a), (b) and (c) are the correlation matrices computed on the 2287, 6860 and 2287 samples from the test set (unseen during training), train set and generated set, respectively. For the synthetic data, the distribution of gene correlations is slightly flatter (see also Supplementary Appendix SD). (d), (e) and (f) are the dendrograms computed obtained by performing hierarchical clustering with complete linkage on the same datasets. Our model closely matches the expression patterns both in terms of correlations and clusters

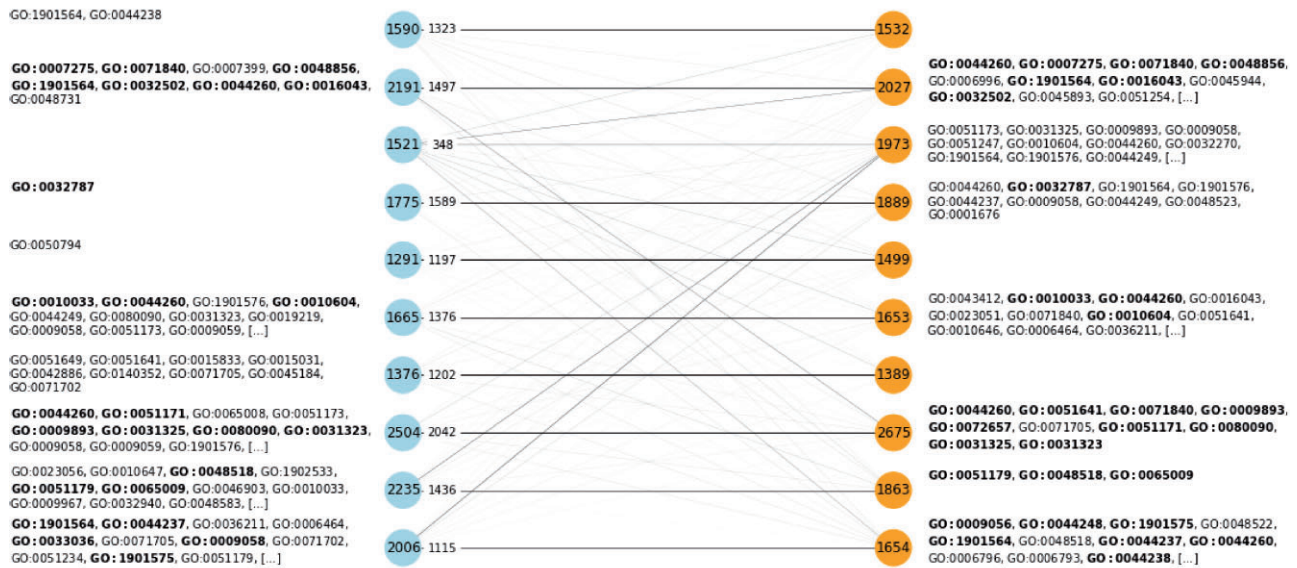


Fig. 3. Cluster analysis on the real and synthetic expression datasets. We performed k -means clustering with $k=10$ clusters both on the test (real) and the generated datasets. Blue and orange nodes represent real and synthetic clusters, respectively. The value of each node corresponds to the number of genes of that cluster. We matched real and synthetic clusters according to the number of shared genes and, for each real cluster, we display as edge labels the number of matching genes for the top association. The width of each edge is proportional to the number of shared genes. We further performed an overrepresentation test using GOfuncR (Grote, 2020) with a family-wise error rate threshold of 0.05. We show the enriched Gene Ontology terms next to the corresponding cluster and highlight in bold those that are common between each top matching pair of clusters (see Supplementary Appendix SF for a detailed list on the enriched Gene Ontology terms). These results suggest that gene clusters and enriched biological processes are similar at a global scale

vector and a set of covariates (e.g. tissue- and cancer-types). If we clamp the latent variable and covariates to a fixed value, we can then use the generator to produce gene expression data for the same *counterfactual* patient with and without cancer. Then, if we observe changes in gene expression, they can only be due to the cancer factor, since all the other latent covariates are fixed. This is something

that cannot be done for the real GTEx + TCGA data because we do not have access to counterfactuals and, therefore, changes in gene expression between healthy and cancer donors might be explained by a large number of confounders in addition to cancer. Other works have explored this idea in the context of image editing (Antipov *et al.*, 2017; Karras *et al.*, 2020; Perarnau *et al.*, 2016).

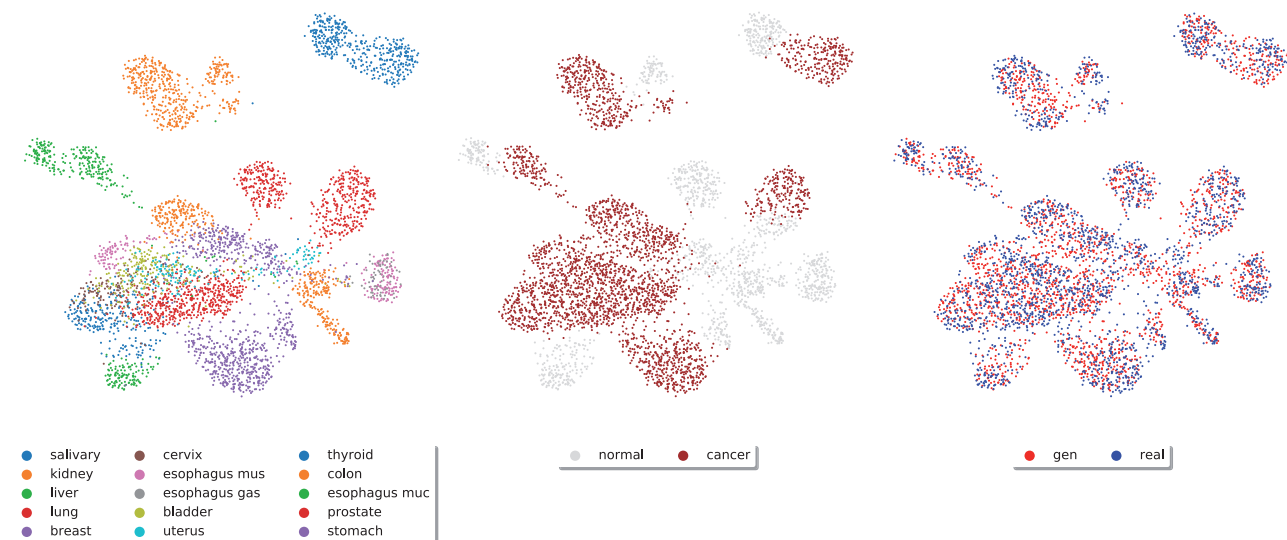


Fig. 4. UMAP representation of RNA-seq data across 15 tissue-types for both normal and cancer, combining data from the test set (2287 samples) and synthetic data (2287 samples). The first plot is coloured by tissues, the second indicates which samples are carcinogenic and the third distinguishes samples between real and synthetic

To rank the genes according to their sensitivity to cancer in our model, we generate pairs of *counterfactual* gene expression values in several tissues. For each pair of measurements, we fix all the latent variables to the same state and generate healthy and cancerous gene expression. Then, we compute the differential expression values and average the results across 1000 runs, obtaining differential gene expression signatures for each cancer-type. Finally, we rank the genes separately for each cancer-type and report the resulting ranking in [Supplementary Appendix SE](#) (along with references from the literature for each reported gene). The results are *causal* in the sense that a change in gene expression can only be due to cancer, since all the other determinants of expression in the model are fixed. Importantly, the gene ranking is sensitive to the ability of our model to estimate the probability distribution of gene expression conditioned on the covariates.

6 Conclusion

In this article, we implemented a simulator based on a WGAN-GP (Gulrajani et al., 2017). We studied the problem of generating realistic transcriptomics data and analysed several statistical properties of gene expression in two case studies: *E.coli* microarray data and human RNA-seq data across a broad range of tissue- and cancer-types.

For the first case study, we compared the ability of our simulator to preserve gene expression properties related to the underlying GRN of the organism, e.g. *E.coli*. Importantly, we noted that two widely used simulators, SynTREN and GNW, poorly preserve correlation properties of gene expression, such as TF-TG and TG-TG correlations. This has important implications on the benchmarks of algorithms that reverse engineer the GRN from transcriptomics data. In particular, if these correlations are not well-preserved, it is not possible to guarantee the generalizability of such algorithms to real data. Conversely, we showed that the data produced by our model is highly realistic according to these metrics, outperforming SynTREN and GNW by a large margin.

For the second case study, we trained our model on a dataset that combines RNA-seq data from the GTEx and TCGA projects. Our analysis showed that the proposed approach preserves correlation and clustering properties, suggesting that the model learns to approximate the gene expression manifold in a biologically meaningful way. Furthermore, our model seems to capture tissue- and cancer-specific properties of transcriptomics data. Finally, we proposed a tool based on the simulator that might be employed by researchers to explore *candidate* cancer driver genes, with potential application in biomarker discovery.

Acknowledgements

We thank Nikola Simidjievski for the helpful comments and discussion.

Funding

The project leading to these results have received funding from ‘la Caixa’ Foundation (ID 100010434), under the agreement LCF/BQ/EU19/11710059.

Conflict of Interest: none declared.

References

- Aguet, F. et al. (2019) The GTEx consortium atlas of genetic regulatory effects across human tissues. *Science*, 369, pp. 1318–1330.
- Antipov, G. et al. (2017) Face aging with conditional generative adversarial networks. *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, pp. 2089–2093.
- Arjovsky, M. et al. (2017) Wasserstein GAN. *arXiv e-Prints*, arXiv: 1701.07875.
- Bailey, M.H. et al. (2018) Comprehensive characterization of cancer driver genes and mutations. *Cell*, 173, 371–385.
- Colbran, L.L. et al. (2019) Inferred divergent gene regulation in archaic hominins reveals potential phenotypic differences. *Nat. Ecol. Evol.*, 3, 1598–1606.
- Cookson, W. et al. (2009) Mapping complex disease traits with global gene expression. *Nat. Rev. Genet.*, 10, 184–194.
- Emilsson, V. et al. (2008) Genetics of gene expression and its effect on disease. *Nature*, 452, 423–428.
- Evans, W.E. and Relling, M.V. (2004) Moving towards individualized medicine with pharmacogenomics. *Nature*, 429, 464–468.
- Faith, J.J. et al. (2008) Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.*, 36, D866–D870.
- Gama-Castro, S. et al. (2016) RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.*, 44, D133–D143.
- Gamazon, E.R. et al.; GTEx Consortium. (2018) Using an atlas of gene regulation across 44 human tissues to inform complex disease- and trait-associated variation. *Nat. Genet.*, 50, 956–967.
- Goodfellow, I. et al. (2014) Generative adversarial nets. In: *Advances in Neural Information Processing Systems 2*. pp. 2672–2680. Montreal, Canada. Curran Associates Inc.
- Grote, S. (2020) GOfuncR: gene ontology enrichment using FUNC. R package version 1.10.0.
- Gulrajani, I. et al. (2017) Improved training of Wasserstein GANs. *CoRR*, abs/1704.00028.

- Irizarry, R.A. *et al.* (2003) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.*, **31**, e15.
- Irrthum, A. *et al.* (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**, e12776.
- Karras, T. *et al.* (2020) Analyzing and improving the image quality of StyleGAN. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 8107–8116.
- Maier, R. *et al.* (2013) A Turing test for artificial expression data. *Bioinformatics*, **29**, 2603–2609.
- Margolin, A.A. *et al.* (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7.
- Marouf, M. *et al.* (2020) Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nat. Commun.*, **11**, 1–12.
- McInnes, L. *et al.* (2020) UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *The Journal of Open Source Software*, **3**, 861.
- Mikolov, T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, Lake Tahoe, USA, pp. 3111–3119.
- Mortazavi, A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nat. Methods*, **5**, 621–628.
- Perarnau, G. *et al.* (2016) Invertible conditional GANs for image editing. *NIPS Workshop on Adversarial Training*.
- Salgado, H. *et al.* (2006) RegulonDB (version 5.0): *Escherichia coli* k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res.*, **34**, D394–D397.
- Schaffter, T. *et al.* (2011) GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, **27**, 2263–2270.
- Schena, M. *et al.* (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.
- Sirota, M. *et al.* (2011) Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Sci. Transl. Med.*, **3**, 96ra77.
- Stegle, O. *et al.* (2012) Using probabilistic estimation of expression residuals (PEER) to obtain increased power and interpretability of gene expression analyses. *Nat. Protoc.*, **7**, 500–507.
- Tieleman, T. and Hinton, G. (2012) Lecture 6.5—rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.*, **4**, 26–31.
- Van den Bulcke, T. *et al.* (2006) SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, **7**, 43.
- Virmaux, A. and Scaman, K. (2018) Lipschitz regularity of deep neural networks: analysis and efficient estimation. In: *Advances in Neural Information Processing Systems*. Montreal, Canada, pp. 3835–3844.
- Wang, Q. *et al.* (2018) Unifying cancer and normal RNA sequencing data from different sources. *Sci. Data*, **5**, 180061.
- Weinstein, J.N. *et al.*; The Cancer Genome Atlas Research Network. (2013) The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.*, **45**, 1113–1120.
- Yu, J. *et al.* (2004) Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, **20**, 3594–3603.