Towards Biologically Plausible and Private Gene Expression Data Generation

Dingfan Chen*
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
dingfan.chen@cispa.de

Marie Oestreich*
German Center for Neurodegenerative Diseases (DZNE)
Bonn, Germany
marie.oestreich@dzne.de

Tejumade Afonja*
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
tejumade.afonja@cispa.de

Raouf Kerkouche
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
raouf.kerkouche@cispa.de

Matthias Becker†
German Center for Neurodegenerative Diseases (DZNE)
Bonn, Germany
matthias.becker@dzne.de

Mario Fritz†
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
fritz@cispa.de

ABSTRACT

Generative models trained with Differential Privacy (DP) are becoming increasingly prominent in the creation of synthetic data for downstream applications. Existing literature, however, primarily focuses on basic benchmarking datasets and tends to report promising results only for elementary metrics and relatively simple data distributions. In this paper, we initiate a systematic analysis of how DP generative models perform in their natural application scenarios, specifically focusing on real-world gene expression data. We conduct a comprehensive analysis of five representative DP generation methods, examining them from various angles, such as downstream utility, statistical properties, and biological plausibility.

Our extensive evaluation illuminates the unique characteristics of each DP generation method, offering critical insights into the strengths and weaknesses of each approach, and uncovering intriguing possibilities for future developments. Perhaps surprisingly, our analysis reveals that most methods are capable of achieving seemingly reasonable downstream utility, according to the standard evaluation metrics considered in existing literature. Nevertheless, we find that none of the DP methods are able to accurately capture the biological characteristics of the real dataset. This observation suggests a potential over-optimistic assessment of current methodologies in this field and underscores a pressing need for future enhancements in model design.

KEYWORDS

gene expression data, differential privacy, data generation, neural networks, synthetic data

* Equal contribution
† Joint last authorship

Proceedings on Privacy Enhancing Technologies 2024(2), 531–554

https://doi.org/10.56553/popets-2024-0062

1 INTRODUCTION

Genomic data is considered a goldmine for medical researchers, enabling them to tackle a wide array of challenges. These challenges range from identifying patients at risk of specific diseases, to developing tailored drugs to enhance treatment reliability and reduce care duration. Gene expression data stands as one of the most extensively utilized forms of genomic data. More specifically, in a cell, the instructions on how to build the cell's proteins are encoded in the DNA as genes. In order to produce proteins, copies of the genes are made from the DNA in the form of messenger RNAs (mRNAs) which are then translated into proteins. The more mRNA copies are made of a gene, the more of the corresponding protein can be produced. Conditions such as environmental stimuli or diseases can alter the kind and quantity of proteins that are being produced. Thus, the cell's response to such conditions is reflected in the transcription of genes, i.e., the strength of their expression. Measuring gene expression has therefore become an essential biomedical tool in order to understand how a cell, tissue or organism responds to the conditions it is exposed to [10, 40].

Nevertheless, the use of gene expression data is not without danger, as it can threaten patient privacy [25]. The precise nature of the information it contains could attract the interest of malicious entities, capable of exploiting it for multiple purposes. For example, an insurance company could choose to raise the coverage cost for a patient predisposed to a serious illness. Additionally, publishing information about a person's genetic predispositions for stigmatized diseases can severely impact their social life and societal acceptance.

In light of these concerns, there arose a need to protect individual privacy and avoid such problems, leading to exploration of methods that are able to generate synthetic data backed by rigorous privacy guarantees. Such approaches involve creating synthetic datasets that reflect the characteristics of real gene expression data while providing strong theoretical differential privacy (DP) guarantees. Nonetheless, employing DP entails introducing randomness during the training process, which inevitably compromises the quality of the produced synthetic data. Furthermore, as we strive for stronger privacy guarantees, the randomness required for privacy increases proportionally, further affecting the quality of the synthetic data to

a larger extent. This underscores the well-known trade-off between privacy and utility.

Despite significant advances in DP data generation methods that report both good generation quality and privacy guarantees, the majority of quality assessment have unfortunately focused solely on downstream utility. A notable gap persists in evaluations that overlook the preservation of essential statistical and biological characteristics. These characteristics are, however, crucial for ensuring the fidelity and applicability of the generated data. In real-world scenarios, the challenge becomes even more pronounced due to the vast feature space inherent in gene expression data, which stands in stark contrast to the often limited number of available samples. Consequently, the effectiveness of existing methods, previously tested primarily on basic benchmark datasets with relatively simple distributions, remains unclear when applied to real-world gene expression data.

In this paper, we fill this gap by presenting the first systematic quality assessment of synthetic gene expression data produced by five benchmark DP generation models with diverse characteristics. Our assessment encompasses five metrics, spanning various aspects from downstream utility, statistical fidelity, to biological plausibility. Our extensive experimental results reveal intriguing findings: (1) significant privacy risks do exist if the generative models are trained non-privately, while DP training (even with a high privacy budget of $\varepsilon = 100$) greatly mitigates such risks; (2) almost all methods manage to achieve seemingly near-perfect performance in terms of standard utility metrics while providing a reasonably strong privacy guarantee (e.g., $\varepsilon \leq 10$), yet none of the DP models succeed in producing biologically plausible data.

In summary, the key contributions of our study are outlined below:

- Our work presents the first comprehensive and systematic analysis of DP generation methods applied to real-world gene expression data. Our extensive investigation encompasses five diverse generation models, five metrics targeting three principal aspects, providing the first comprehensive view for the current state of real-world applicability of DP generation methods.
- Our analysis reveals crucial insights, highlighting the limitations of existing evaluations that predominantly focus on a single aspect, namely, downstream utility. In contrast, our thorough assessment establishes a reliable evaluation framework that effectively addresses the misconceptions arising from these one-dimensional evaluations.
- Our compelling findings, complemented by an in-depth discussion, offer fresh perspectives for the future development in the related field. With our systematic assessment, we aim to steer DP generation methods towards improved practicality in real-world applications involving sensitive data.

2 RELATED WORK

2.1 Models for Synthetic Gene Expression Data

Various types of generative models have been employed for generating synthetic gene expression data. Variational autoencoders and deep Boltzmann machines have been used to generate data that aids in designing studies and planning analysis for large experiments [39]. Generative adversarial networks have been exploited for generating gene expression data to combat the challenges of low sample sizes via data augmentation, which is specifically motivated by the unfavorable ratio of samples to features in these datasets [18, 23]. Additionally, synthetic gene expression data has also been used to train imputation methods for handling missing data [27]. However, none of these methods ensure privacy during the whole data generation process. Given that genome-related data, including the gene expression data, is highly privacy-sensitive [25], applying existing works in real-world scenarios becomes challenging due to privacy regulations.

To the best of our knowledge, there is a lack of research delving into the differentially private generation of synthetic gene expression data. While some studies, like [37], have investigated the private generation of synthetic data within the realm of medical data at large, a dedicated focus on gene expression data remains notably absent.

2.2 Measuring Quality of Synthetic Gene Expression Data

A variety of methods have been applied in the past to assess the quality of synthetic gene expression data from a biological standpoint. These methods have been used both in the context of bulk as well as single-cell RNA-seq data. Bulk RNA-sequencing refers to the process of sequencing the mRNA transcripts from a sample containing a collection of many cells [20]. The resulting data thus reports the average expression strength of each gene across these cells. Single-cell RNA-sequencing on the other hand, first separates the cells present in the sample before sequencing each individually, generating an expression profile at cell resolution rather than sample resolution [20, 36]. The methods used for evaluating this data comprise the comparison of expression data distributions [6, 39, 43] by looking at mean and median expressions, proportion of zero counts (in single-cell cases) and coefficients of variation. Also, metrics related to functional biology have been applied [18, 23, 27, 39**?**], including preservation of gene-gene correlations, gene ontology terms, differentially expressed genes and clusters in reduced dimensional space, using for example t-SNE, PCA, UMAP or after feature selection.

3 PRELIMINARIES

3.1 Threat Model

The objective of an adversary is to infer private information about individuals in the training datasets by launching various privacy attacks, such as membership inference attack (MIA), which aims to ascertain if a particular data point was used in training the dataset.

We consider two common scenarios for synthetic data generation from an attack standpoint:

- A trained generator generates the synthetic data (e.g., Section 4.1-4.4). In this case, the adversary can have either black-box access or white-box access to the generator. Black-box access means the adversary can only access the synthetic data generated by querying the model through an API. White-box access allows the adversary to access the generator's internal state, including its parameters.

- The synthetic data is directly generated without using any generator (e.g., Section 4.5). In this scenario, the adversary only has access to the synthetic data.

While our privacy model protects against the most powerful adversaries, as discussed below in Section 3.2, our experiments consider the scenario with the most knowledgeable adversary who has white-box access to the trained generator, as well as the practical scenario where only the synthetic data is accessible.

3.2 Privacy Model

We aim to develop a solution that protects against potential attacks as delineated in our threat model in Section 3.1. Specifically, we adopt differential privacy (DP), which ensures the difficulty to infer the presence of any record in the training dataset, even when the adversary has white-/black-box access to the trained generator and/or to the synthetic data. As a result, any potential negative impact on an individual's privacy cannot be attributed to their involvement in the training phase (up to ε and δ). For instance, if an insurance company accesses the generator or the synthetic data (from DP generation methods) and decides to increase an individual's insurance premium, such a decision cannot be attributed to the individual's data presented in the training dataset.

Definition 3.1 ((ε, δ)-DP [12])**.** A randomized mechanism $\mathcal{M}$ with range $\mathcal{R}$ is (ε, δ)-DP, if

$$\Pr[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(\mathcal{D}') \in O] + \delta$$

holds for any subset of outputs $O \subseteq \mathcal{R}$ and for any adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, where $\mathcal{D}$ and $\mathcal{D}'$ differ from each other by adding or removing one training example, i.e., $\mathcal{D}' = \mathcal{D} \cup \{x\}$ or $\mathcal{D} = \mathcal{D}' \cup \{x\}$ for a data sample x. The privacy parameter ε is the upper bound of privacy loss, and δ is the probability of breaching DP constraints. Smaller values of both ε and δ translate to stronger DP guarantees and better privacy protection.

Definition 3.2 (Gaussian Mechanism [12])**.** Let $f : X \to \mathbb{R}^d$ be an arbitrary d-dimensional function with L_2-(global) sensitivity Δ_f^2:

$$\Delta_f^2 = \max_{\mathcal{D}, \mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \tag{1}$$

The Gaussian Mechanism $\mathcal{M}_\sigma$, parameterized by σ, adds noise into the output, i.e.,

$$\mathcal{M}_\sigma(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \boldsymbol{I}). \tag{2}$$

$\mathcal{M}_\sigma$ is (ε, δ)-DP for $\sigma \geq \sqrt{2 \ln (1.25/\delta)} \Delta_f^2 / \varepsilon$.

Theorem 3.1 (Post-processing Theorem [12])**.** If $\mathcal{M}$ satisfies (ε, δ)-DP, $F \circ \mathcal{M}$ will satisfy (ε, δ)-DP for any data-independent function F with $\circ$ denoting the composition operator.

The post-processing theorem guarantees that if a DP generation model is (ε, δ)-DP, releasing the trained generator and the synthetic dataset will also be privacy-preserving, with the privacy cost bounded by ε and δ.

3.3 Biological Criteria

Differential Expression. When diseases and other pathological conditions affect the body, they can alter gene activation within cells, contributing to the manifestation of symptoms. The specific set of genes whose expression levels vary from one disease to another are commonly referred to as *differentially expressed (DE) genes*. Identifying DE genes that distinguish between two conditions is a fundamental step in gene expression analysis [4, 11, 30, 31]. Differential expression can occur as either *up-regulation* or *down-regulation*, meaning that the expression of genes is significantly *increased* or *decreased* in one condition compared to another, respectively (see Section 5.3 for the formal definition).

Gene Co-Expression. Genes that are involved in the same biological pathways often form a functional group or *module*, meaning they collectively respond to a condition by similar changes in the expression strength. For example, all genes involved in fighting off a bacterial infection will be activated together when such a pathogen enters the body. Such genes are referred to as *co-expressed*. In order to identify activated or inactivated biological pathways, detecting such modules of co-expressed genes is a common step in the analysis of gene-expression data [19, 26]. Specifically, co-expression between a pair of genes with indices j and k is quantified using their *Pearson correlation coefficient* r_{jk} with

$$r_{jk} = \frac{\sum_{i=1}^{n} (x_j^{(i)} - \bar{x}_j)(x_k^{(i)} - \bar{x}_k)}{\sqrt{\sum_{i=1}^{n} (x_j^{(i)} - \bar{x}_j)^2} \sqrt{\sum_{i=1}^{n} (x_k^{(i)} - \bar{x}_k)^2}}, \tag{3}$$

where $x_j^{(i)}$ and $x_k^{(i)}$ are the expression values of genes j and k in sample i, respectively, while $\bar{x}_j$ and $\bar{x}_k$ are the mean expression values of the two genes across n biological samples. Groups of genes with high Pearson correlation coefficients are considered *modules* of co-expressed genes, with $r_{jk} > 0.7$ are typically considered as biologically significant co-expressions.

4 MODELS

Given the real dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^n$ consisting of n samples $(\boldsymbol{x}^{(i)}, y^{(i)})$ with $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{1, ..., C\}$ denoting the features and class labels respectively, the objective of the generation methods is to capture the real underlying distribution $p(\boldsymbol{x}, y)$ and generate synthetic data samples $(\tilde{\boldsymbol{x}}, \tilde{y})$ that mimic the statistical characteristics of the real samples from $\mathcal{D}$. In our case, the feature vector $\boldsymbol{x}^{(i)}$ represents the gene expression level and the class label $y^{(i)}$ corresponds to the disease type, with d and C denoting the feature dimension and number of label classes, respectively.

In this work, we explore the most prominent categories of (DP) generation methods found in the literature: (1) *density estimation (probability distribution fitting)*, (2) *graphical models-based* methods, (3) *marginal-based* methods, and (4) *deep generative models*. A summary of these methods and their diverse characteristics can be found in Table 1.

Method	Category	Attribute type	DP sanitization
RON-Gauss	Density estimation	continuous only	one-shot
VAE	Deep generative model	continuous	iterative
GAN	Deep generative model	continuous	iterative
Private-PGM	Graphical model	discrete only	one-shot
PrivSyn	Marginal	discrete only	one-shot

Table 1: Summary of Models.

4.1 RON-Gauss

RON-Gauss [7] generates synthetic data by drawing samples from a multivariate Gaussian distribution fitted in a projected space of the real data. Specifically, it operates by executing the following steps: Firstly, the data is pre-processed to ensure it possesses bounded sensitivity and adheres to the regularity conditions for the Diaconis-Freedman-Meckes effect (which guarantees the data will exhibit Gaussian-like distribution after projection with high probability). Next, a random orthonormal (RON) projection is applied on the pre-processed data, i.e., $\overline{\boldsymbol{X}} = \boldsymbol{W}^T\boldsymbol{X}$ with $\boldsymbol{W} \in \mathbb{R}^{d\times p}$ signifying the RON projection matrix and $\boldsymbol{X}$ representing the pre-processed data matrix. Subsequently, a multivariate Gaussian model is fitted onto the projected data. During the inference stage, new samples are drawn from the fitted Gaussian distribution and are inversely projected into the original data space to form synthetic data samples. To maintain privacy, DP noise is added into both the mean and covariance of the fitted Gaussian distribution. Moreover, the Gaussian model is independently applied to each label class to facilitate label-conditional generation, which aligns with the concept of a Gaussian mixture model (GMM), where each label class forms a mode of the GMM. The detailed algorithm is presented in Algorithm 1.

Algorithm 1: RON-Gauss

Input: Dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, projection dimension p, noise scale σ
Output: Synthetic dataset $\mathcal{S}$
for c **in** $\{1, ..., C\}$ **do**
(1) Extract samples with label class c to form data matrix $\boldsymbol{X}_c \in \mathbb{R}^{d\times n_c}$;
(2) Pre-process data and compute the mean:
- Pre-normalize:$\boldsymbol{x}^{(i)} := \boldsymbol{x}^{(i)}/\|\boldsymbol{x}^{(i)}\|_2 \quad \forall \boldsymbol{x}^{(i)} \in \boldsymbol{X}_c$
- Compute the DP mean: $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\boldsymbol{x}^{(i)} + \mathcal{N}(0, \sigma^2\boldsymbol{I})$
- Center the data: $\boldsymbol{x}^{(i)} := \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_c \quad \forall \boldsymbol{x}^{(i)} \in \boldsymbol{X}_c$
- Re-normalize: $\boldsymbol{x}^{(i)} := \boldsymbol{x}^{(i)}/\|\boldsymbol{x}^{(i)}\|_2 \quad \forall \boldsymbol{x}^{(i)} \in \boldsymbol{X}_c$

(3) Apply RON projection: $\overline{\boldsymbol{X}}_c := \boldsymbol{W}^T\boldsymbol{X}_c \in \mathbb{R}^{p\times n_c}$;
(4) Derive the DP covariance: $\Sigma_c = \frac{1}{n_c}\overline{\boldsymbol{X}}_c\overline{\boldsymbol{X}}_c^T + \mathcal{N}(0, \sigma^2\boldsymbol{I})$;
(5) Synthesize data for class c by drawing samples from the Gaussian distribution $\widetilde{\boldsymbol{x}}^{(i)} \sim \mathcal{N}(\boldsymbol{W}^T\boldsymbol{\mu}_c, \Sigma_c)$;
(6) Inversely project and recenter: $\widetilde{\boldsymbol{x}}^{(i)} := \boldsymbol{W}\widetilde{\boldsymbol{x}}^{(i)} + \boldsymbol{\mu}_c$ and construct the synthetic set $\mathcal{S}_c = \{(\widetilde{\boldsymbol{x}}^{(i)}, c)\}_{i=1}^{n_c}$;
end
return Synthetic dataset $\mathcal{S} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_C$

4.2 VAE

The Variational Autoencoder (VAE) [17] is a type of deep generative model that consists of both an encoder and a decoder. During training, these two components are cascaded and optimized to reconstruct data under pre-defined similarity metrics such as L_1/L_2 loss. The encoder (denoted as q_ϕ) maps input data $\boldsymbol{x}$ into a latent space, while the decoder (denoted as p_θ) maps the encoded latent representation back into the data space. Meanwhile, VAE regularizes the encoder by imposing a prior P_z over the latent code distribution. This regularization encourages the latent code to form a simple distribution that is amenable to sampling. During inference, new latent codes $\boldsymbol{z}$ are sampled from the prior distribution P_z and then fed into the decoder to generated synthetic samples. The formal VAE objective is composed of a reconstruction term and a prior regularization term:

$$\min_{\theta,\phi} \mathcal{L}_{VAE} = -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[p_\theta(\boldsymbol{x}|\boldsymbol{z})] + KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|P_z) \quad (4)$$

where $KL(\cdot\|\cdot)$ denotes the KL divergence, $\boldsymbol{z}$ and $\boldsymbol{x}$ stand for the latent code and the real data, respectively. $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ represents the probabilistic encoder parameterized by ϕ, and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ represents the probabilistic decoder parameterized by θ. In practice, the prior P_z is always chosen to be a unimodal Gaussian distribution and $\boldsymbol{z}$ is sampled using the reparameterization trick, facilitating a closed-form derivation of the second term.

We employ the class conditional (CVAE) [32] for label-conditional generation. In this framework, both the encoder and the decoder receive additional (one-hot) label information y. Formally, the training objective can be expressed as:

$$\min_{\theta,\phi} \mathcal{L}_{CVAE} = -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},y)}[p_\theta(\boldsymbol{x}|\boldsymbol{z},y)] + KL(q_\phi(\boldsymbol{z}|\boldsymbol{x},y)\|P_z) \quad (5)$$

During the generation process, labels are generated based on their occurrence rates in the real dataset. Privacy constraints is incorporated in the training stage by replacing the regular stochastic gradient descent (SGD) update with DP-SGD [1], which involves clipping the per-example gradients and adding calibrated random noise to the mini-batch gradients.

4.3 GAN

The Generative Adversarial Network (GAN) [14] is another widely used type of deep generative model. It comprises two neural network components, a generator G_θ and a discriminator D_ϕ, which are trained simultaneously in an adversarial manner. The generator takes random noise $\boldsymbol{z}$ (latent code) as input and generates samples that approximate the distribution of the training data. Conversely, the discriminator evaluates both generator-generated samples and real training data samples, aiming to distinguish between the two sources. Throughout training, these two modules engage in a competitive process, each adapting to the other: the generator seeks to generate progressively more realistic samples to deceive the discriminator, while the discriminator learns to distinguish the two sources more accurately. The standard GAN training objective can be formulated as

$$\min_{\theta}\max_{\phi} \mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}}[\log(D_\phi(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim P_z}[\log(1 - D_\phi(G_\theta(\boldsymbol{z})))] \quad (6)$$

where θ, ϕ denote the parameters of the generator and the discriminator respectively. P_{data} stands for the real data distribution, and the P_z is the prior distribution of the latent code. The first term in the objective prompts the discriminator to output high scores for real data samples. In contrast, the second term encourages the discriminator to assign lower scores to generated samples, while the generator is optimized to maximize the discriminator's output score. During inference, the generator will receive new latent code samples z drawn from the known prior distribution P_z, often standard Gaussian, and produce synthetic data samples.

For private training, we adopt the DP Wasserstein GAN (DP-WGAN) [3] implementation and its conditional variant to integrate label information during generation. Specifically, the Wasserstein

distance [5] is used as the training objective with the label information acting as auxiliary input for both the generator and discriminator:

$$\min_{\theta}\max_{\phi}\mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}}[D_\phi(\boldsymbol{x},y)]-\mathbb{E}_{\boldsymbol{z}\sim P_z}[D_\phi(G_\theta(\boldsymbol{z},y),y)] \quad (7)$$

The DP guarantee is ensured by employing DP-SGD for discriminator updates, which in turn guarantee the privacy of the whole GAN model and the synthetic data due to the post-processing theorem (Theorem 3.1).

4.4 Private-PGM

The Private Probabilistic Graphical Models (Private-PGM) framework [24] is designed to construct undirected graphical models from DP noisy measurements over low-dimensional marginals, which facilitates the generation of new synthetic samples via sampling from the learned graphical model. Specifically, Private-PGM operates on records consisting of discrete attributes. Formally, a record is denoted as $\boldsymbol{x} = (x_1, ..., x_d, x_{d+1})$ where each feature attribute x_i for all $i\in\{1, ..., d\}$ and the label $y = x_{d+1}$ fall within a discrete finite domain. Let C represent a collection of *measurement sets*, where each $C\in C$ is a subset of $\{1, ..., d + 1\}$ (i.e., the combinations of attributes), and let $\boldsymbol{v}_C$ define the marginal probability vector on C. Private-PGM first obtains DP noisy measurements $\boldsymbol{m}_C = \boldsymbol{Q}_C\boldsymbol{v}_C + \mathcal{N}(0, \sigma_C^2\boldsymbol{I})$ with $\boldsymbol{Q}_C$ denoting the linear marginal query set over measurement set C and $\mathcal{N}(0, \sigma_C^2\boldsymbol{I})$ representing the noise introduced by the Gaussian mechanism (with σ_C the noise scale determined by the desired privacy level ε_C and δ. Refer to Definition 3.2). Subsequently, it estimates the marginal $\hat{\boldsymbol{v}}$ that best explain all the noisy measurement $\hat{\boldsymbol{v}} = \arg\min_{\boldsymbol{v}} \|\boldsymbol{Q}\boldsymbol{v} - \boldsymbol{m}\|$ where $\boldsymbol{Q}$ is a block-diagonal matrix with diagonal blocks $\{\boldsymbol{Q}_C\}_{C\in C}$ (i.e., combining all the query set $\boldsymbol{Q}_C$) and $\boldsymbol{m} = (\boldsymbol{m}_C)_{C\in C}$ the combined vector of measurements. Meanwhile, it estimates the parameter of the graphical model using existing graph inference and learning algorithms such as belief propagation on a junction tree.

In general, $\boldsymbol{Q}_C$ can represent a complex set of linear queries expressed over C, and its selection can be adaptively tailored to downstream objectives. In our work, we adhere to the default implementationwhere $\boldsymbol{Q}_C$ is set to be an identity matrix. This configuration renders the measurement $\boldsymbol{m}_C$ equivalent to the corresponding noisy marginal $\boldsymbol{v}_C + \mathcal{N}(0, \sigma_C^2\boldsymbol{I})$. Moreover, for computational feasibility, we adopt the basic configuration offered by the official implementation that sets $C = \big\{\{1\}, ..., \{d+1\}\big\} \cup \big\{\{1, d+1\}, ..., \{d, d+1\}\big\}$, which encompasses all one-way marginals as well as the 2-way marginals associated with the label attribute. The privacy budget is allocated uniformly across each measurement, i.e., $\varepsilon_C = \varepsilon/|C|$ with ε the total privacy cost due to sequential composition.

4.5 PrivSyn

Similar to Private-PGM, PrivSyn [44] operates on data with discrete attributes to obtain measurable (noisy) marginals. However, while Private-PGM explicitly constructs factorized sparse graphical models, PrivSyn directly generates data from the noisy marginal measurements. This approach inherently allows the use of an implicitly dense graphical model, enhancing its expressiveness capacity.

PrivSyn is structured to execute the following steps sequentially:

- *Marginal selection*: This step selects the most informative marginals from the candidate set to optimize the privacy-utility trade-off.
- *Noise addition*: DP noise is added to the selected marginal measurements, ensuring privacy guarantee.
- *Post-processing*: This phase ensures consistency from the noisy measurements. It addresses issues such as negative marginal measurements, cases where probabilities do not sum up to 1, and aligning different marginals that share common attributes.
- *Data Synthesis*: Starting with a randomly initialized synthetic dataset, this step iteratively updates it to ensure alignment with the marginal measurements.

In our experimental evaluation, we omit the more involved 2-way marginal selection step for our dataset, as this step is prohibited by the significant computation and privacy costs, which scale quadratically with the feature dimensions. Instead, we utilize all 2-way marginals linked with the label attribute, aligning with the approach taken in Private-PGM to ensure a fair comparison. Apart from this, we adhere to the default configuration of the official implementation, which allocates the privacy budget at a ratio of 1 : 8 between publishing the 1-way and 2-way marginals.

5 MULTI-DIMENSIONAL EVALUATION OF SYNTHETIC GENE EXPRESSION DATA

Our study delved into a comprehensive assessment of various models. This evaluation was executed through a meticulous analysis of model performance across three main aspects: **utility** (Section 5.1), **statistical** (Section 5.2), and **biological** (Section 5.3) evaluation. Each aspect encompasses distinct metrics: *machine learning efficacy* for **utility** evaluation, *marginal* (*histogram intersection*) and *joint* (*distance to closest record*) closeness for **statistical** evaluation, as well as *differential expression* and *gene co-expression* for **biological** evaluation.

5.1 Utility Evaluation

5.1.1 Machine Learning Efficacy. Evaluating the quality of synthetic data typically involves a standard procedure of assessing its performance within a downstream task. This evaluation determines whether the synthetic data, when used as a replacement for the real data, can accomplish the desired task with comparable effectiveness. This is executed by training machine learning models on real (train) data and evaluating their performance on held-out (test) data. Subsequently, a parallel model is trained on synthetic data and evaluated using the same held-out data. The choice of evaluation metrics is determined by the specific nature of the task at hand. In our work, we adopt the standard *accuracy* score for evaluating the disease classification task.

5.2 Statistical Evaluation

Utility-based metrics, however, often offer an incomplete perspective due to their narrow evaluation lens, presenting a single facet of the model's performance, which can occasionally lead to misleading impressions. In order to address this potential bias, it becomes crucial to incorporate additional statistical metrics that emphasize the fidelity of the generation process. This entails assessing how effectively the model captures both the marginal distribution and

the underlying joint distribution of the data, providing a more comprehensive understanding of its performance.

5.2.1 Histogram Intersection. The *histogram intersection* serves as a prevalent qualitative tool for visualizing one-dimensional data (i.e., single columns/attributes), enabling a comprehensive exploration of the data's distribution characteristics. Understanding such single-dimensional distributions can be pivotal for subsequent preprocessing and analysis steps. Prior studies have harnessed this metric to compare the distributions of synthetic and real data by selecting specific attributes from the real dataset and overlaying the histograms of the corresponding real data onto the synthetic ones. This technique, referred to as *distribution matching plots*, provides a qualitative assessment of how closely the two distributions align.

However, relying solely on qualitative measures has its limitations, particularly when confronted with large feature sets like gene expression data. Manually visualizing each column becomes impractical. This necessitates a quantitative approach that maintains a similar essence but can be aggregated to yield a single score. The normalized *histogram intersection metric* proposed in [2] is applicable in such scenario. It is computed as the sum of the minimum probability values between the real data column and the synthetic data column. This sum is subsequently averaged across the various columns in the dataset (see Equation 9). In contrast to other analogous techniques like the *Wasserstein distance* [16] or *Jensen-Shannon divergence score* [21], the *histogram intersection* score demonstrates superior performance and exhibits a strong correlation with other metrics [2] (see Appendix Fig. 10)[1]. This quantitative approach strikes a balance between comprehensiveness and practicality, making it an effective tool for evaluating the quality of the generated data.

$$p_c = \frac{s_c}{|\mathcal{D}|\Delta_i} \quad q_c = \frac{t_c}{|\mathcal{S}|\Delta_i} \tag{8}$$

$$\mathrm{HI}(\boldsymbol{p}_i, \boldsymbol{q}_i) = \sum_c \min(p_c, q_c) \tag{9}$$

$$\text{Overlap Score} = \frac{1}{d}\sum_i \mathrm{HI}(\boldsymbol{p}_i, \boldsymbol{q}_i) \tag{10}$$

where $\mathbf{p}_i$ and $\mathbf{q}_i$ denote the histogram representations of the probability distributions for the real ($\mathcal{D}$) and synthetic ($\mathcal{S}$) datasets within feature i, respectively. The terms p_c and q_c represent the proportions of category c for feature i, with s_c/t_c denoting the counts of real/synthetic samples in category c. The factor Δ_i is introduced as a normalization term, specifying the bin size for numerical features. The term $\mathrm{HI}(\mathbf{p}_i, \mathbf{q}_i)$ represents the histogram intersection score for feature i. The dimensionality of the feature space is denoted by d. The *Overlap Score* is computed by averaging the histogram intersection scores across all features.

5.2.2 Distance to Closest Record. The *distance to closest record* metric aims to measure the similarity between the *joint* distribution of real and synthetic data. Obtaining an exact measurement of the joint distribution is inherently challenging and always infeasible, as the underlying probability distribution of the real data is unknown and generally intractable. To circumvent this, we approximate the alignment of joint distributions using k-nearest neighbors (KNN). This involves computing the Euclidean distance between each synthetic data sample and its k nearest neighbors in either the held-out or training set. The objective is to evaluate the plausibility of each synthetic sample being real. The final KNN Distance score is the average across all synthetic dataset samples and various k values, as defined in Equation 12.

$$d_k(\tilde{\boldsymbol{x}}) = \mathrm{first}_k\Big(\mathrm{sort}\Big(\big\{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \mid \forall \boldsymbol{x} \in \mathcal{D}_{\text{train/test}}\big\}\Big)\Big) \tag{11}$$

$$\text{KNN Distance Score} = \frac{1}{|\mathcal{S}| \cdot k}\sum_{\tilde{\boldsymbol{x}} \in \mathcal{S}}\sum_{i=1}^{k} d_{k,i}(\tilde{\boldsymbol{x}}) \tag{12}$$

where $\mathcal{S}$ denotes the synthetic set, $\mathcal{D}$ the real dataset, $d_k(\tilde{\boldsymbol{x}})$ is a sequence contain the k smallest values of distances (where $\mathrm{sort}(\cdot)$ represents the sorting operation in ascending order and $\mathrm{first}_k(\cdot)$ denotes the operation for retrieving the first k elements from the sorted sequence), with $d_{k,i}(\tilde{\boldsymbol{x}})$ denoting the i-th element of $d_k(\tilde{\boldsymbol{x}})$.

5.3 Biological Evaluation

5.3.1 Differential Expression. There are several methods to measure differential expression, but many of them make strong assumptions on the distribution underlying gene expression data [4, 22, 31]. However, the question of which distribution gene expression data follows has been subject to debate for many years [**?**]. To avoid making any (potentially false) assumptions regarding the distribution, we chose a non-parametric test for the identification of differentially expressed genes, namely the *Wilcoxon signed rank test* [42]. For each pair of conditions in the data, the test was conducted on the expression values of each gene measured across the samples of the respective condition. We ran the test using the pairwise Wilcoxon function from the R-package *scran* (version 1.26.2) and using the alternative hypothesis for each side to differentiate between *up-* and *down-regulation*. We considered a gene as differentially expressed between two conditions if the p-value was at most 0.05. The reconstruction of DE-genes by different generative models M at varying privacy levels ε is quantified via the mean true positive rate (*TPR*) defined as follows.

$$\mathrm{TPR}_{\mathcal{M},\varepsilon} = \frac{\sum_{\{a_i,a_j\}\subset\mathcal{A},a_i\neq a_j}\left(\mathrm{TPR}^{\text{up}}_{a_i,a_j,\mathcal{M},\varepsilon} + \mathrm{TPR}^{\text{down}}_{a_i,a_j,\mathcal{M},\varepsilon}\right)}{2\cdot\binom{|\mathcal{A}|}{2}} \tag{13}$$

where $\mathcal{A}$ denotes the set of condition pairs (each pair representing different disease types distinguished by unique label classes in our case). $\mathrm{TPR}^{\text{up (down)}}_{a_i,a_j,\mathcal{M},\varepsilon}$ signifies the true positive rate for identifying *up-regulated* (or *down-regulated*) DE-genes within synthetic data generated by the model $\mathcal{M}$ under a given privacy budget ε, in comparison to the actual DE-genes observed in the real dataset for conditions a_i and a_j. $\binom{|\mathcal{A}|}{2}$ represents the count of all possible unordered condition pairs.

5.3.2 Gene Co-Expression. To assess if groups of co-expressed genes that are present in the real data were preserved in the synthetic data, we applied *hCoCena* [26], an R-package that enables the integration of different gene expression datasets, i.e., the real and the synthetic data in our case, and their subsequent joint co-expression analysis. The tool creates a gene co-expression network for each set, which is a weighted graph $G = (V, E)$, where the nodes V represent

[1] The histogram intersection metric defined here also corresponds to 1 - total variation distance, a popular metric that quantifies the similarity between two probability distributions.

Class	AML	ALL	CML	CLL	Other	**Total**
# Samples	508	12	14	13	634	**1181**

Table 2: Dataset summary. Listed are the different sample classes present in the dataset and the number of samples in each class.

genes, edges E represent co-expressions and the edges are weighted with the co-expression strength. The weight w is computed as the *Pearson correlation coefficient* r (see Equation 3) between their expression values across samples, such that $w(e_{j,k})=r_{jk}$. Afterwards, genes that are not significantly strongly co-expressed according to a user-defined correlation cut-off with any other gene are discarded to only include strong co-expressions that are potentially biologically meaningful. A gene co-expression network is created for each dataset. We then used these co-expression networks to identify the number of co-expressions (i.e., graph edges) that were correctly reconstructed in the synthetic data and the number of spurious co-expressions introduced in the synthetic data that did not exist in the real data. Additionally, modules of strongly co-expressed genes were identified in the network of the real dataset using the *Leiden community detection algorithm*. We investigated the **mean group fold-changes (GFCs)** for the detected modules across conditions in the real and the synthetic data. GFCs are a metric for the average expression of a module in a group of samples, i.e. all samples of a particular experimental condition, essentially representing the activation or deactivation of the module under the given condition.

6 EVALUATION

6.1 Dataset

The generative models were trained on a bulk RNA-seq dataset compiled by Warnat-Herresthal *et al.* [41], which comprises a reasonable number of independent samples, rendering it suitable for DP training (see Section 8 for detailed discussion). The dataset is structured as a matrix, with rows corresponding to *samples* and columns to *features*. Each row represents a biological specimen obtained from a patient, while each column indicates the expression level of a particular gene. The expression levels are quantified by RNA-seq counts, with higher integer values indicating greater gene activity. It comprises samples from 5 disease classes, 4 classes of which are types of leukemia and the fifth class is the category "*Other*", which is made up of samples from various other diseases as well as healthy controls. The 4 leukemia types are acute myeloid leukemia ("*AML*"), acute lymphocytic leukemia ("*ALL*"), chronic myeloid leukemia ("*CML*") and chronic lymphocytic leukemia ("*CLL*"). Sample counts per class are listed in Table 2. As per the original publication, the data were normalized with DeSeq2 [22] to account for varying sequencing depths and RNA composition, which is necessary to compare expression levels of different samples and conduct a DE-gene analysis. Given the high dimensionality of the features (more than 12k genes) and the comparatively low sample size (1181), we reduced the feature space to 958 genes. Notably, even this reduced feature dimension remains significantly high, especially when compared with standard benchmarking datasets which typically comprise merely dozens of features.

These 958 genes were not selected randomly but based on their characterization as *landmark genes* in the LINCS L1000 project [35]. The landmark genes were identified as representative genes that, when measured, allow the inference of around 20k other genes.

Pre-processing and Post-processing. In accordance with standard practices, we pre-processed our data prior to model training. For `RON-Gauss`, `GAN`, and `VAE`, which operate on continuous data expected to be well-centered, we standardized each feature by subtracting its mean and dividing by its standard deviation. Conversely, for `Private-PGM` and `PrivSyn` which rely on discrete representations for computing marginals, we discretized each feature into four bins based on its quantiles: <25%, 25%-50%, 50%-75%, and >75%. This approach was chosen to accurately represent *up-* and *down-regulation*, while also maintaining a condensed format (resulting in a limited number of bins after discretization) for an optimized privacy-utility trade-off.

After training and generation, we implemented the following post-processing measures:

- For `RON-Gauss`, `GAN`, and `VAE`: We reverted the standardization by multiplying the generated data features by the standard deviation and adding back the mean (both the standard deviation and the mean were pre-computed on the real dataset).
- For `Private-PGM` and `PrivSyn`: We mapped the generated discrete data back to the original continuous mean value associated with each bin.

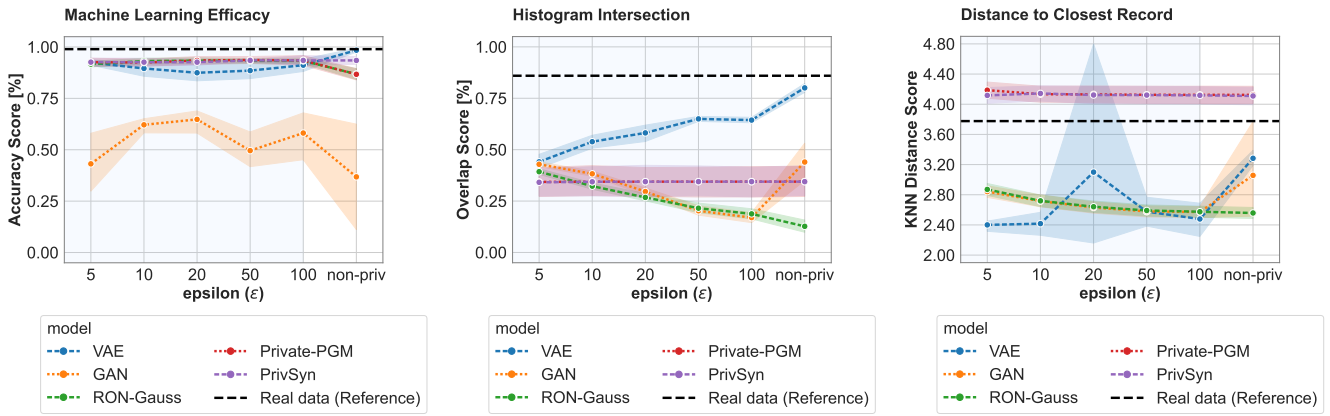
We verify the efficacy of our approach via our preliminary experiments: the continuous pre- and post-precessing was proved to be lossless, while the discrete one did *not* affect the biological and utility evaluation.

In line with the common evaluation protocol adopted in DP literature, we do *not* incorporate DP into the pre- and post-processing process, and the label class occurrence ratio is treated as public information and used during generation. This approach aids in producing meaningful evaluation results and offers a more accurate indication of performance, particularly given our challenging setup. However, it is crucial to note that in real-world applications, all such processes including the hyperparameter selection [28] would require DP sanitization to ensure stringent privacy protection. Although implementing such sanitization is generally technically straightforward (e.g., either computed on public data or using DP techniques such as Algorithm 2 in [38] and [13] for DP sanitization techniques applicable to continuous and discrete processing, respectively), it can lead to considerable utility loss in bio-data, mainly due to limited sample sizes, which warrants further discussion and investigation.

6.2 Setup

We follow the official implementation for methods that offer open-source code: `RON-Gauss`[2], `GAN`[3], `Private-PGM`[4], `PrivSyn`[5] and adopt the default hyperparameter setting tuned for general tabular datasets. We adhere to such setting as further attempts at fine-tuning did not give rise to notably better results in our preliminary experiments.

[2] https://github.com/inspire-group/RON-Gauss/tree/master
[3] https://github.com/nesl/nist_differential_privacy_synthetic_data_challenge/
[4] https://github.com/ryan112358/private-pgm
[5] https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms/DPSyn



(a) (b) (c)

Figure 1: Utility Evaluation by Machine Learning Efficacy, and Statistical Evaluation by Histogram Intersection and Distance to Closest Record. Shown in (a) are the Accuracy Scores for the *Machine Learning Efficacy* metric across 5 various models for the DP-case (blue shading) with varying ε values, alongside the non-private case. Similarly, (b) and (c) display the Overlap Score and K-Nearest Neighbors Distance Score for the *Histogram Intersection* metric and *Distance to Closest Record* metric, respectively. Evaluations encompassed two seeds for training split creation and two synthetic dataset randomizations. The presented values represent means across these randomization seeds. The black dashed line represents the reference score on actual train-test data, signifying the best attainable score.

We use the RDP accountant implementation from TensorFlow privacy[6] for `VAE` and `GAN`. For the `VAE` model, which lacks an official DP implementation, we tuned the key hyperparameters (including the weight of the reconstruction loss term, the number of training iterations, the gradient clipping bound, the batch size, and the latent dimension) via grid-search. We repeat the experiments over different random seeds and report the mean and standard deviation over these seeds by default. For biological evaluation where results from different seed cannot be aggregated, we detail the outcomes for each individual seed separately. The δ is set to be 10^{-5} by default across our experiments.

7 EXPERIMENTS

We study five different generative models: `VAE`, `GAN`, `RON-Gauss`, `Private-PGM`, and `PrivSyn`, which encompass diverse categories, attribute types, and DP sanitation approaches, as summarized in Table 1. Our assessment was conducted under two scenarios: initially, without the imposition of DP constraints, and subsequently, with DP integration using values of epsilon (ε) ranging from 5 to 100, signifying a spectrum from high to low privacy levels. We did not reduce the privacy budget to values smaller than 5, as the models fail to achieve reasonable results at this threshold. For models such as `VAE` and `GAN` that were not originally designed with DP protections, we incorporate DP to the gradients following the DP-SGD framework [1] to create their respective private variants. Conversely, for inherently privacy-centric models like `RON-Gauss`, `Private-PGM`, and `PrivSyn`, we set the noise scale to be zero to simulate their non-private counterparts. These diverse models were then evaluated using the metrics detailed in Section 5. We set the real data as *Reference* (See the dashed black lines in Fig. 1), which represents the score of each metric when applied to the real training data and then evaluated on the real held-out (i.e., test) data.

7.1 Utility Evaluation

For the Machine Learning Efficacy metric, given the classification nature of the task (predicting diverse disease types using gene expression data), we employ a widely-used and straightforward machine learning approach known as logistic regression. This model undergoes training as outlined in Section 5.1.1. The chosen evaluation metric is the *accuracy* score.

Results and Findings. The outcomes, depicted in Fig. 1(a), portray the machine learning utility scores for various generative models across differing privacy levels, ranging from high ($\varepsilon = 5$) to low ($\varepsilon = 100$). In the non-private context (termed as non-priv in Fig. 1(a)), we observe that all five models—with the exception of `GAN`—exhibit a substantial utility score ranging from 86% to 98%. This shows a moderate decrease of 0.5% to 12% relative to the reference point set by real data (black dashed line). Within the private realm ($\varepsilon = 5, 10, 20, 50, 100$), models such as `Private-PGM`, `PrivSyn`, and `RON-Gauss` display consistent high utility, encountering a reduction of less than 7.4% in very high privacy conditions ($\varepsilon = 5$) to a 5.8% drop in situations with lower privacy ($\varepsilon = 100$). Notably, these models demonstrate a higher utility as ε increases. Remarkably, the utility metric easily saturates, even with a simple probabilistic model (i.e., the unimodal Gaussian as in `RON-Gauss`), while the `VAE` exhibits slight advantages in the non-private case. The `GAN` model generally performs worse in terms of the utility metric and exhibits relatively high variance, potentially due to the unstable nature of its adversarial training process, which is exacerbated in our dataset with limited samples.

[6] https://github.com/tensorflow/privacy/blob/master/research/hyperparameters_2022/rdp_accountant.py

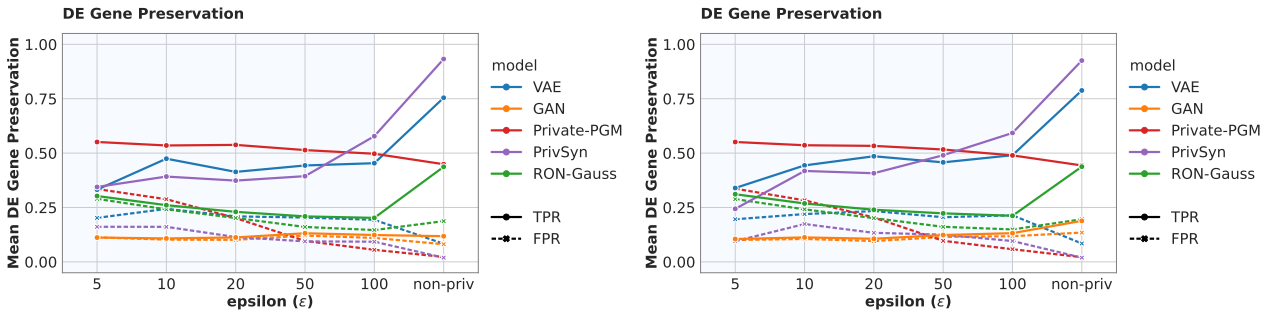


Figure 2: Biological Evaluation by DE-Gene Preservation. Shown is the preservation of DE-genes (true positive rate (TPR): solid lines; false positive rate (FPR): dashed lines) across the tested models for the DP-case (indicated by blue shading) with different values of ε and the non-private case. The evaluation was performed for two different seeds used for creating the training split (left and right plot). The presented values are means across two different seeds set for generating the data (except for `Private-PGM` and `PrivSyn`, where seeding is not possible).

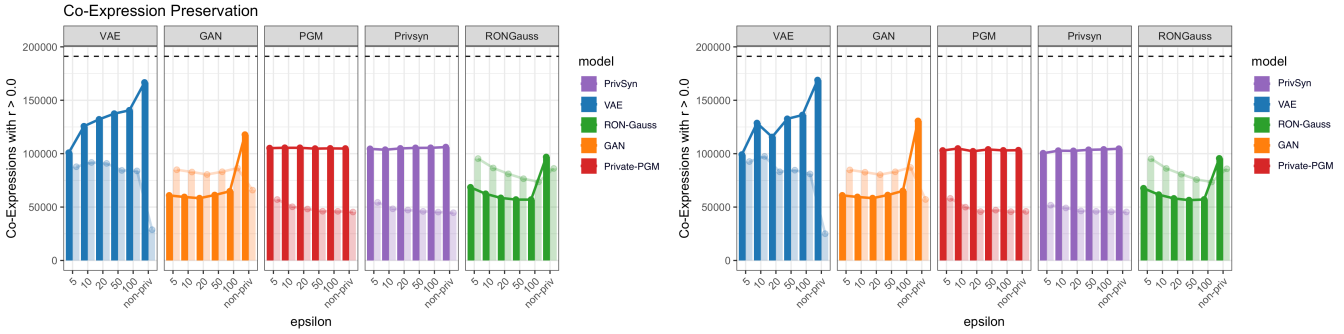


Figure 3: Biological Evaluation by Co-Expression Preservation for $r > 0$. Shown is the co-expression preservation across the tested models for different values of ε as well as the non-private case for two different seeds used for creating the training split (left and right plot). Specifically, non-transparent bars give the number of correctly reconstructed co-expressions with Pearson Correlation Coefficient $r > 0$ and an associated p-value < 0.05, while semi-transparent bars give the number of co-expressions introduced by the model that did not exist in the real data. The dashed black line indicates the number of co-expressions in the real data. All values shown are means across two different seeds set for generating the data (except for `Private-PGM` and `PrivSyn`, where seeding is not possible).

7.2 Statistical Evaluation

7.2.1 Histogram Intersection. We initiate by subjecting the numerical column to min-max pre-processing, a technique that rescales values to fit within the range of 0 to 1. Following this normalization, a discretization binning process is employed, utilizing 25, 50, and 100 bin size, which provides an approximated representation of the numerical column's distribution, and thus ensures tractability. No additional pre-processing steps are required for the discrete and categorical columns. Our computation of the *Overlap Score* adheres to the definition in Equation 10.

Results and Findings. Fig.1(b) illustrates the overlap score, which serves as the mean of the histogram intersection scores between the columns of real and synthetic data, as detailed in Section 5.2.1. In general, across both private and non-private cases, most models exhibit subpar performance on this metric. An exception stands out: the `VAE` model (depicted by the blue line). Remarkably, for the non-private case, it showcases an impressive overlap score of ~80%, experiencing only a 6.8% relative drop compared to the reference set by the real data (black dashed line). This performance trend consistently improves from the high privacy ($\varepsilon = 5$) to the low privacy case ($\varepsilon = 100$), indicating that the synthetic data's marginal distribution increasingly resembles that of the real data, with the relaxation of privacy constraints. However, this does not uniformly apply to all models. For instance, the `RON-Gauss` model (represented by the green line) shows an unexpected behavior—its overlap score is higher in the very high privacy case ($\varepsilon = 5$) compared to the non-private case, exhibiting an 85% drop in performance relative to the real data reference. This outcome is surprising given that this model involves continuous attribute types, which should typically lead to a moderately increasing overlap score as ε increases. Similarly, the `GAN` model follows a similar trend to `RON-Gauss`, but it demonstrates a higher overlap score in the non-private case, with a reduced relative drop of 48.8%. We conjecture that such seemingly abnormal behavior may be partially explained by the fact that both `GAN` and `RON-Gauss` did not capture the marginal distributions faithfully even in the non-private case, which making the results more influenced by randomness than by the learnability. The inferior performance of the `Private-PGM` and `PrivSyn` models in this metric

can potentially be attributed to the loss in precision resulting from the reverse transformation inherent in the discretization process, which may dominate the additional information loss incurred by privacy constraints. Interestingly, despite the modest performance in this metric, the same models excel in the private case for the machine learning efficacy metric. This underlines the necessity of evaluating synthetic data from various generative models across an array of metrics to gain a comprehensive understanding of their behavior relative to real data.

7.2.2 Distance to Closest Record. This metric aims to approximate the likelihood that a synthetic data sample originates from the distribution of real data samples. This measurement relies on the K-Nearest Neighbors (KNN) approximation technique. In our experimental setup, we specifically set the value of k to 10, which dictates the computation of the KNN Distance Score according to Equation 12. We use the scikit-learn KNN estimator[7] to compute the nearest neighbors distance of each synthetic sample to the real test data. Fig. 1(c) shows the averaged 10-NN distance score for different epsilon values (x-axis) and diverse generative models. A higher proximity of this score to the reference established by the real data implies a greater likelihood that the *joint* distribution of real and synthetic data aligns closely. Scores falling below the reference point set by real data imply that the synthetic data samples are closely aligned with the distribution of the real test data. However, it is essential to exercise caution while interpreting these results due to the relatively small size of the test set. Making assertive conclusions based solely on these findings might be premature.

Results and Findings. Intuitively, we anticipate that the score for this metric should be lower, indicating closer alignment to the real data reference (depicted by the black dashed line) in the non-private setting. As privacy levels increase, we expect a moderate increase in the distance—moving from $\varepsilon = 100$ to $\varepsilon = 5$. This examination aims to substantiate the assertions made by prior study [29] that this metric has the potential to quantify privacy. However, the results illustrated in Fig. 1(c) present a counter-intuitive observation. All models, excluding the graphical-based models `Private-PGM` and `PrivSyn`, demonstrate distances below the real data reference. This holds true for both private and non-private scenarios. Notably, the `VAE` model stands out, exhibiting a low distance to the closest test record (i.e closest to the real data reference but still falls below the black dashed line). This shows a relative drop of 48% when contrasted with the reference established by real data. Notably, the `Private-PGM` and `PrivSyn` models, which yield unsatisfactory outcomes in the histogram intersection metric, also exhibit the most substantial distances to the real data reference. This persistent distance above the black dashed line further indicates that the reverse discretization process could lead to a loss of precision in these models. Additionally, for the `VAE` model, across the $\varepsilon = 5$ to $\varepsilon = 50$ range, there's a pronounced variance in scores across different experimental random seeds. This variance might offer insights into the model's sensitivity behavior in the private case.

7.2.3 Summary of Utility and Statistical Evaluation. The observed results of Fig. 1 underscore the necessity of assessing diverse metrics when evaluating synthetic data. Moreover, it brings to light an intriguing revelation: even if the synthetic data strays from both marginal and joint distributions, it still exhibits the capacity to maintain substantial downstream utility tasks. This observation reinforces the significance of a comprehensive evaluation approach that considers various aspects of data behavior and performance.

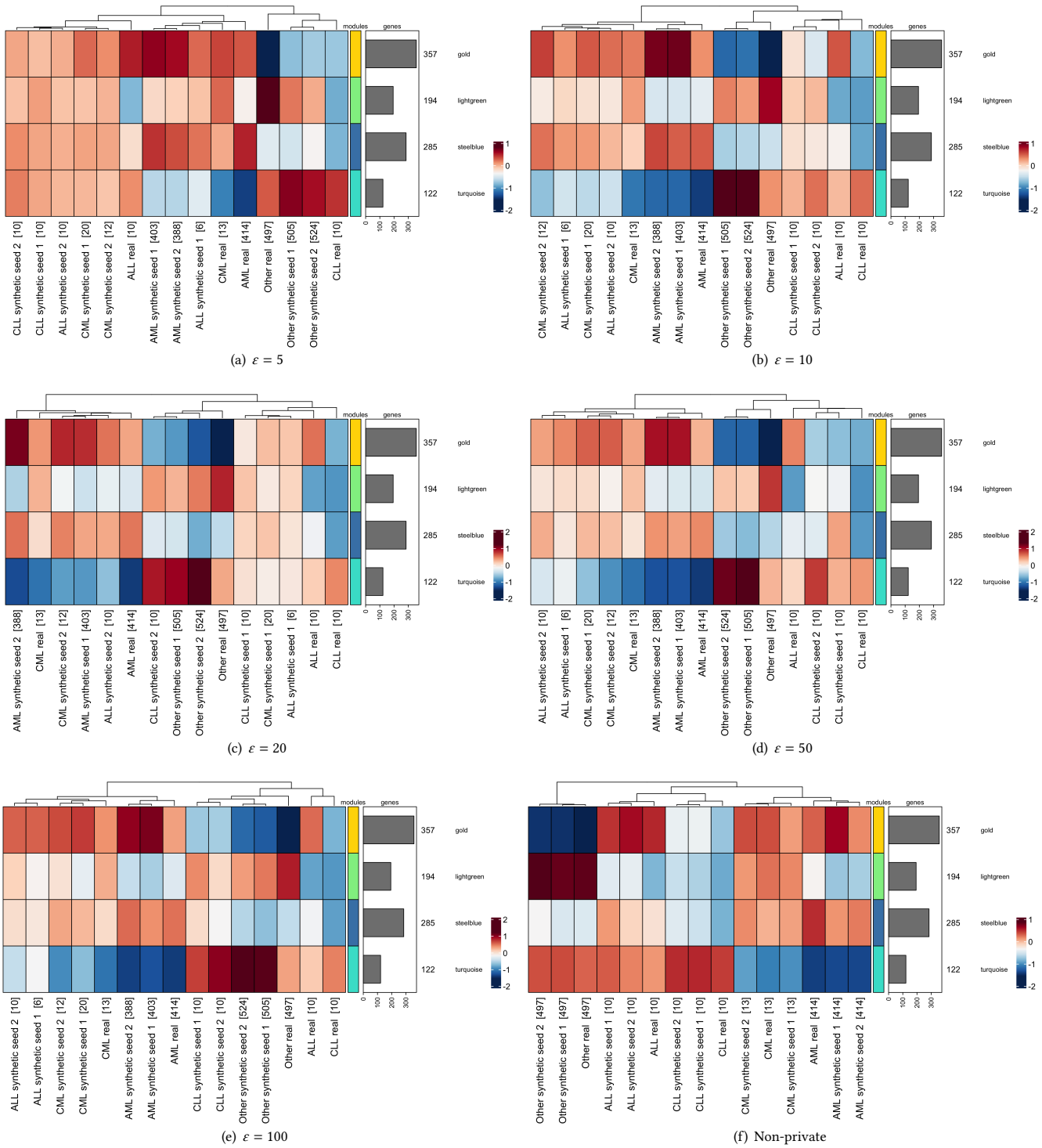
7.3 Biological Evaluation

To evaluate the different models for biological soundness, we assessed their capabilities of maintaining two biological aspects in the generated synthetic data: (1) the preservation of differential expression by assessing the TPR and FPR of reconstructed DE-genes per model and across privacy parameters and (2) the preservation of co-expressions between genes, i.e., their Pearson Correlation Coefficients r as well as the activation of co-expressed modules. The models were evaluated once without the constraint of DP and then with DP using ε = 5, 10, 20, 50, 100.

7.3.1 Differential Expression. We first compared the models' ability to maintain DE-genes in a **non-private** case. As shown in Fig. 2, it can be observed that the TPR was high for `PrivSyn` and `VAE` models, reaching more than 75% on both data split seeds. `RON-Gauss` , `Private-PGM` and `GAN` showed subpar results, with the `GAN` model performing particularly poorly. Regarding the FPR, all models maintained rates below 25%, with `PrivSyn` and `Private-PGM` reaching FPRs close to zero. For the **DP**-case, we observe from Fig. 2 the following:

- `VAE`: At a privacy parameter $\varepsilon = 100$, the TPR decreases noticeably in comparison to the non-DP setting from around 75% on average to approximately 50%. As ε is reduced further, the TPR continues to show a decreasing tendency, albeit at a less steep rate. Even at the lowest privacy budget of $\varepsilon = 5$, the TPR of `VAE` remains higher than that of the `GAN` in the non-DP setting. Moreover, `VAE` shows better or equal TPR than `PrivSyn` at low ε values, and outperforms `RON-Gauss` across all ε but underperforms `Private-PGM` once DP is introduced.
  The FPR increases slightly when introducing DP but remains largely stable for different values of ε.
- `GAN`:The TPR of DE-genes in the `GAN` model observed under non-DP conditions remains poor at the introduction of DP and decreasing ε, staying below 20%, while the FPR remains stable (around 10%) across all ε.
- `RON-Gauss`: The TPR of the `RON-Gauss` model drops when introducing DP. Intriguingly, and somewhat against expectations, it exhibits a slight improvement as the privacy loss ε decreases, yet it still only attains low values (around 30%). Concurrently, the FPR steadily increases with decreasing ε, eventually approaching the TPR.
- `Private-PGM`: The TPR of `Private-PGM` exhibits a slight increase with decreasing ε, with `Private-PGM` outperforming all other models for $\varepsilon \leq 50$. Conversely, the FPR rate also increases drastically, reaching around 35%.
- `PrivSyn`: While `PrivSyn` showed near perfect TPR in the non-DP setting, it is strongly impacted by the introduction of DP, falling below `VAE` and `Private-PGM` for $\varepsilon \leq 50$. This performance loss is similarly reflected in the increasing FPR with decreasing ε values.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html



(a) $\varepsilon = 5$ (b) $\varepsilon = 10$

(c) $\varepsilon = 20$ (d) $\varepsilon = 50$

(e) $\varepsilon = 100$ (f) Non-private

Figure 4: Activation patterns of co-expressed gene modules in VAE for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (*rows*) in the real and the synthetic data sampled with two different seeds. The dendrograms representing the hierarchical clustering of the sample groups differentiated by label class and seed, with each *column* corresponding to a distinct group. Optimally, samples with the same label classes should be adjacent, indicating that they are clustered together. Numbers on the right indicate the number of genes per module, numbers in square brackets on the bottom indicate the number of samples per condition and dataset. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation.

In summary, while PrivSyn and VAE demonstrate good preservation of DE genes in the non-DP case, their performances drop when introducing DP and they are surpassed by the Private-PGM model. However, this boost in DE-gene preservation of the Private-PGM model is accompanied by an increasing false positive rate, possibly indicating that the Private-PGM model generally tends to generate more DE genes, however without biological correctness. Both the GAN and the RON-Gauss models perform poorly on this metric, especially in the DP case.

7.3.2 Co-expression. Here, we investigated both the general preservation of co-expressed genes as well as the activation and deactivation of strongly co-expressed gene sets, so-called *modules*, detected in the real data. The preserved co-expressions as well as the activation patterns of co-expressed gene modules were assessed once for all positive correlations identified in the data (r > 0) (Fig. 3, Appendix Fig. 11-15) and once after filtering for only highly co-expressed genes (r > 0.7) (Appendix Fig. 16-18). The latter is motivated by the typical interest in strongly correlated genes during co-expression analyses. The detection of gene modules was performed on the real data for these respective filtering thresholds. In both cases, co-expressions were filtered for associated p-values < 0.05.

We first investigate the **non-private** case. When considering all co-expressions with $r > 0$, the VAE reconstructed most of the them while only introducing few false ones that did not exist in the real data (Fig. 3). It further maintained highly similar patterns of *up-* and *down-regulation* in the gene modules (Appendix Fig. 11), with samples clustering by class rather than dataset. The GAN model had less correctly and more incorrectly reconstructed co-expressions than the VAE (Fig. 3) and the patterns of activation in the gene modules do not match the real data (Appendix Fig. 12). The Private-PGM and PrivSyn models had very similar performances, with more correctly than incorrectly reconstructed co-expressions, however only reconstructing half of the co-expressions found in the real data (Fig. 3). The activation of the gene modules was well reconstructed (Appendix Fig. 13, 14). The number of correctly and incorrectly reconstructed co-expressions was almost equal for the RON-Gauss model (Fig. 3) and activation patterns in the gene modules were almost entirely lost (Appendix Fig. 15). Reducing the co-expressions to only those with $r > 0.7$, only the VAE model and one of the sampling seeds for the GAN yielded any results. The VAE correctly reconstructed most co-expressions from the real set but additionally introduced an almost equal number of incorrect co-expressions (Appendix Fig. 16). Activation patterns in gene modules were well preserved (Appendix Fig. 17). In the data generated by the GAN, the number of incorrectly introduced co-expressions was very high (Appendix Fig. 16) and activation patterns in the gene modules remained poor (Appendix Fig. 18).

For the **DP**-case, we list below our findings:

- VAE: For all co-expressions with $r > 0$, the number of correct co-expressions reconstructed by the VAE reduced gradually when introducing DP with decreasing ε (Fig. 3). Meanwhile, the number of incorrect co-expressions more than doubled. Activation patterns of gene modules were well maintained for the classes *CML*, *AML* and *Other* at $\varepsilon = 100$ and 50. For lower ε, the characteristic patterns of the modules were increasingly lost as illustrated in Fig. 4, indicated by the increasing lack of distinctive colors. While the order of gene modules (rows) is fixed to improve comparability, the order of sample groups (columns) is dictated by their hierarchical clustering. This is intended, since it illustrates similarity between module expression of different conditions in the different datasets. In the case of biologically high-quality synthetic data, synthetic samples are expected to co-locate with real samples of the same condition. Note that the results are only shown for one seed used for splitting the dataset for training. Detailed illustrations for all ε and seeds can be found in Appendix Fig. 11. If the synthetic data successfully captured the co-expression modules, disease classes are expected to cluster together across synthetic and real data. Focusing only on highly co-expressed genes with $r > 0.7$, a high number of co-expressions is introduced that do not occur in the real data (Appendix Fig. 16). Preservation of module activation is comparable to that observed when selecting co-expressions with $r > 0$ (Appendix Fig. 17).
- GAN: When considering all co-expressions with $r > 0$, the number of correctly reconstructed co-expressions decreased and the number of incorrect ones increased when introducing DP with $\varepsilon = 100$ and reducing this value did not impact the metric further (Fig. 3). The module activation patterns from the real data are almost entirely lost with the modules demonstrating homogeneous activation (Appendix Fig. 12). When filtering for $r > 0.7$ there were no co-expressions left for any of the ε-values.
- Private-PGM & PrivSyn: The Private-PGM and PrivSyn models demonstrated similar behavior, with the number of reconstructed co-expressions barely being affected by introducing varying levels of privacy in comparison to the non-DP case (Fig. 3). Private-PGM maintained the module activation patters for very high ε-values (100 and 50) and for lower ε (20, 10, 5) patterns of large classes such as *AML* and *Other* where maintained, but degraded for the smaller classes (Appendix Fig. 13). Similar results are observed for PrivSyn, with the exception that the degradation of activation patterns already starts at $\varepsilon = 50$ (Appendix Fig. 14). Like the GAN, both models did not generate any significant co-expressions exceeding $r > 0.7$.
- RON-Gauss: While in the non-DP case, the number of incorrect co-expressions was still slightly lower than that of correct ones, this changes in the DP-case (Fig. 3). Decreasing values of ε, however, not only gradually increased the number of incorrectly reconstructed ones, but also that of the correctly reconstructed co-expressions. The gene modules lose their distinctive patterns, showing uniform activation and thus the synthetic data is clustering distinctly away from the real data for all ε (Appendix Fig. 15). As was the case for all models but the VAE, no high co-expressions with $r > 0.7$ were generated by the RON-Gauss model.

In summary, all models except the VAE struggled at correctly recreating strong co-expressions and even the VAE was prone to introducing a high number of incorrect co-expressions for $r > 0.7$. Also for weaker co-expressions, introducing DP strongly impaired the utility of the data both in terms of general co-expressions as well as the activation and inactivation of highly co-expressed modules, with only high ε-values of 100 and 50 maintaining the co-expression structure in the data of some models but not offering any considerable privacy.

8 DISCUSSION AND FUTURE DIRECTIONS

Private vs. non-private synthetic data. The biological evaluation of the different models yielded that some model types are capable of generating synthetic data with high biological utility in the non-DP case. However, the incorporation of DP, though essential for maintaining privacy, significantly hampers their performance. In examining the generally top-performing `VAE` models through membership inference attacks (Section 3.1), we found that non-DP training poses a considerable privacy risk, with AUC-ROC scores of 0.949 and 0.614 for white-box (implemented following [15]) and black-box attacks (implemented following [9]), respectively. Notably, setting the privacy budget at a relative high level of ε=100 resulted in a rapid decline of AUC-ROC scores to around 0.52 in both scenarios. While such high privacy budgets ε =100/50 in some cases still allowed good reconstruction of biology properties as measured by our metrics, these budgets are generally too high to be considered strictly privacy-preserving.

Challenges of low sample regime. As has become apparent in the analysis of activation patterns of co-expressed modules, classes with low sample counts were the first to lose their activation patterns with decreasing privacy budgets. However, such low sample sizes are highly common in gene expression datasets given the often low availability of sampling material. This is particularly the case for rare diseases or samples that can only be acquired with invasive and/or risky medical procedures. Another point that requires addressing is the feature space. The results presented here were achieved on a strongly reduced feature space of approximately 1000 genes, with gene expression datasets often comprising 20-times as many features. The observed limitations of differentially private data generation can thus be expected to increase further when attempting to generate full sets.

Comparing models. The biological evaluation indicated that some model architectures (`VAE`, `PrivSyn` and sometimes `Private-PGM`) are better than others (`GAN`, `RON-Gauss`) at learning and generating such highly complex, non-normally distributed data like gene expressions. In general, `VAE` stands out with the best overall performance, likely because of their substantial expressive capacity, which outperforms simpler probabilistic models like `RON-Gauss` and methods dependent on low-dimensional approximations, such as `PrivSyn` and `Private-PGM`. Moreover, VAEs benefit from stable training processes, advantageous in scenarios with limited samples, unlike the less stable GANs. However, incorporating privacy into this process presents challenges, while maintaining biological utility in a privacy-preserving manner requires further research and possibly more data.

Dependent data. In certain scenarios where the dataset used contains dependent records—such as those associated with the same individual (e.g., single-cell data), a transition to a more advanced level of protection becomes imperative, wherein the goal shifts to preserving each group of dependent records (referred to as Group-level DP). However, this elevation in privacy protection comes with the trade-off of injecting more noise, potentially leading to a greater compromise in the quality of the synthetic data. Furthermore, the task of defining a set of dependent records is not always straightforward. For instance, while it is evident that individuals within the same family often share a common genomic heritage, the extent of relatedness to consider when forming such groups remains ambiguous. Determining whether to include only immediate family members like parents and siblings or to encompass more distant relatives poses an additional challenge. Due to these intricate aspects of privacy considerations, we opt to exclude single-cell datasets from our analysis, despite their potential size advantage for assessing non-DP generative models.

General-purpose synthetic data vs. task-specific data. Providing general-purpose private synthetic data that is useful for all kinds of downstream tasks while preserving statistical and biological properties is still a highly challenging task. Having accurate generators would also imply a strong model and insights for the respective domain, which is often not the case for many bio-medical applications. In addition, small sub-population might not be represented and suffer from mode collapse issues of the generator. It has also been recently questions to what extend such an ultimate solution can be achieved at all [33, 34]. While it is difficult to predict how these trade-off develop in the future, the increased available of such medical data will have a positive effect. In addition, task-specific data generation (e.g. [8]) in a data distillation approach can relax the objectives, but is also departing from the goal of preserving statistic and biological properties by mostly focusing on downstream utility.

9 CONCLUSIONS

We provide the first systematic analysis of non-private and differentially private generation of gene expression data that covers five diverse modeling approaches ranging from simple density estimation over graphical models to deep generative models. Our analysis encompasses a diverse set of metrics that shed light on the quality of the generated data in terms of statistical and biological properties as well as down-stream utility. A key message of our work is that such a broad evaluation is necessary in order to understand the limitations of current generators. Overall, simple estimators fall behind in performance but equally very complex models like GAN are suffering from the low sample regime as typically encountered in bio-medical applications. While downstream utility can be strong, the synthetic data itself might not retain statistical nor biological properties. Adding privacy preserving estimation and learning of the generators amplifies these problems. A general model recommendation is difficult to provide, as these trade-offs will shift as more data is going to become available in the future. However, we see a tendency that the evaluated graphical models have retained better the differential expression and the variational autoencoder retained better the co-expression - in particular when privacy is added. We release our setup and evaluation framework in order to further drive progress in this domain[8].

ACKNOWLEDGMENTS

This work is supported by the Helmholtz Association within the project Protecting Genetic Data with Synthetic Cohorts from Deep Generative Models (PRO-GENE-GEN), grant No. ZT-I-PF-5-23 and Bundesministeriums fur Bildung und Forschung (PriSyn), grant

[8] https://github.com/MarieOestreich/PRO-GENE-GEN

No. 16KISAO29K. Additionally, this work is partially funded by ELSA – European Lighthouse on Secure and Safe AI funded by the European Union under grant agreement No. 101070617. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them. Moreover, Dingfan Chen was partially supported by Qualcomm Innovation Fellowship Europe.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Tejumade Afonja, Dingfan Chen, and Mario Fritz. 2023. MargCTGAN: A "Marginally" Better CTGAN for the Low Sample Regime. *arXiv preprint arXiv:2307.07997* (2023).

[3] Moustafa Alzantot and Mani Srivastava. 2019. Differential Privacy Synthetic Data Generation using WGANs. https://github.com/nesl/nist_differential_privacy_synthetic_data_challenge/

[4] Simon Anders and Wolfgang Huber. 2010. Differential expression analysis for sequence count data. *Genome Biology* 11, 10 (Oct 2010), R106. https://doi.org/10.1186/gb-2010-11-10-r106

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[6] Alemu Takele Assefa, Jo Vandesompele, and Olivier Thas. 2020. SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. *Bioinformatics* 36, 10 (May 2020), 3276–3278. https://doi.org/10.1093/bioinformatics/btaa105

[7] Thee Chanyaswad, Changchang Liu, and Prateek Mittal. 2019. RON-Gauss: Enhancing Utility in Non-Interactive Private Data Release. *Proceedings on Privacy Enhancing Technologies* 1 (2019), 26–46.

[8] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. 2022. Private Set Generation with Discriminative Information. In *Neural Information Processing Systems (NeurIPS)* (2022-12-01). https://arxiv.org/abs/2211.04446https://arxiv.org/pdf/2211.04446.pdf

[9] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security (CCS)*. 343–362.

[10] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, and et al. 2016. A survey of best practices for RNA-seq data analysis. *Genome Biology* 17, 1 (Jan 2016), 13. https://doi.org/10.1186/s13059-016-0881-8

[11] Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. 2017. RNA-Seq differential expression analysis: An extended review and a software tool. *Plos One* 12, 12 (Dec 2017), e0190152. https://doi.org/10.1371/journal.pone.0190152

[12] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014).

[13] Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. 2021. Differentially Private Quantiles. In *Proceedings of the 38th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 3713–3722.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[15] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. 2019. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Proc. Priv. Enhancing Technol.* 2019, 4 (2019), 232–249.

[16] Leonid V Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management science* 6, 4 (1960), 366–422.

[17] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations (ICLR)*.

[18] Snehalika Lall, Sumanta Ray, and Sanghamitra Bandyopadhyay. 2022. LSH-GAN enables in-silico generation of cells for small sample high dimensional scRNA-seq data. *Communications Biology* 5, 1 (Jun 2022), 577. https://doi.org/10.1038/s42003-022-03473-y

[19] Peter Langfelder and Steve Horvath. 2008. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* 9 (Dec 2008), 559. https://doi.org/10.1186/1471-2105-9-559

[20] Xinmin Li and Cun-Yu Wang. 2021. From bulk, single-cell to spatial RNA sequencing. *International journal of oral science* 13, 1 (Nov 2021), 36. https://doi.org/10.1038/s41368-021-00146-0

[21] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* 37 (1991), 145–151. https://api.semanticscholar.org/CorpusID:12121632

[22] Michael I Love, Wolfgang Huber, and Simon Anders. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15, 12 (2014), 550. https://doi.org/10.1186/s13059-014-0550-8

[23] Mohamed Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S Magruder, Christian F Krebs, and Stefan Bonn. 2020. Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. 11 (Jan 2020), 166. https://doi.org/10.1038/s41467-019-14018-z

[24] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*. PMLR, 4435–4444.

[25] Marie Oestreich, Dingfan Chen, Joachim L Schultze, Mario Fritz, and Matthias Becker. 2021. Privacy considerations for sharing genomics data. *EXCLI journal* 20 (Jul 2021), 1243–1260. https://doi.org/10.17179/excli2021-4002

[26] Marie Oestreich, Lisa Holsten, Shobhit Agrawal, Kilian Dahm, Philipp Koch, Han Jin, Matthias Becker, and Thomas Ulas. 2022. hCoCena: horizontal integration and analysis of transcriptomics datasets. 38 (Oct 2022), 4727–4734. https://doi.org/10.1093/bioinformatics/btac589

[27] Diksha Pandey and Perumal P Onkara. 2023. Improved downstream functional analysis of single-cell RNA-sequence data using DGAN. *Scientific Reports* 13, 1 (Jan 2023), 1618. https://doi.org/10.1038/s41598-023-28952-y

[28] Nicolas Papernot and Thomas Steinke. 2021. Hyperparameter Tuning with Renyi Differential Privacy. In *International Conference on Learning Representations (ICLR)*.

[29] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384* (2018).

[30] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. 43 (Apr 2015), e47. https://doi.org/10.1093/nar/gkv007

[31] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 1 (Jan 2010), 139–140. https://doi.org/10.1093/bioinformatics/btp616

[32] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).

[33] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic Data – Anonymisation Groundhog Day. In *USENIX Security Symposium*.

[34] Theresa Stadler and Carmela Troncoso. 2022. Why the search for a privacy-preserving data sharing mechanism is failing. *Nature Computational Science* 2, 4 (2022), 208–210.

[35] Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, and et al. 2017. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 171, 6 (Nov 2017), 1437–1452.e17. https://doi.org/10.1016/j.cell.2017.10.049

[36] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, and et al. 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* 6, 5 (May 2009), 377–382. https://doi.org/10.1038/nmeth.1315

[37] Amirsina Torfi, Edward A Fox, and Chandan K Reddy. 2022. Differentially private synthetic medical data generation using convolutional GANs. *Information Sciences* 586 (2022), 485–500.

[38] Florian Tramer and Dan Boneh. 2020. Differentially Private Learning Needs Better Features (or Much More Data). In *International Conference on Learning Representations (ICLR)*.

[39] Martin Treppner, Adrián Salas-Bastos, Moritz Hess, Stefan Lenz, Tanja Vogel, and Harald Binder. 2021. Synthetic single cell RNA sequencing data from small pilot studies using deep generative models. *Scientific Reports* 11, 1 (Apr 2021), 9403. https://doi.org/10.1038/s41598-021-88875-4

[40] Zhong Wang, Mark Gerstein, and Michael Snyder. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews. Genetics* 10, 1 (Jan 2009), 57–63. https://doi.org/10.1038/nrg2484

[41] Stefanie Warnat-Herresthal, Konstantinos Perrakis, Bernd Taschler, Matthias Becker, Kevin Baßler, Marc Beyer, Patrick Günther, Jonas Schulte-Schrepping, Lea Seep, Kathrin Klee, and et al. 2020. Scalable Prediction of Acute Myeloid Leukemia Using High-Dimensional Machine Learning and Blood Transcriptomics. *iScience* 23, 1 (Jan 2020), 100780. https://doi.org/10.1016/j.isci.2019.100780

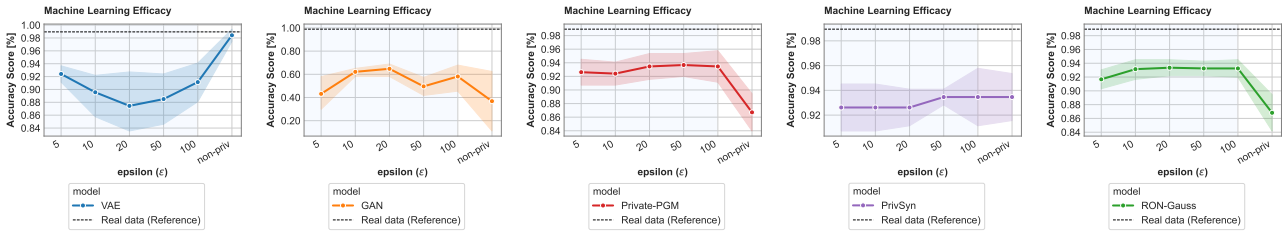
[42] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1, 6 (Dec 1945), 80. https://doi.org/10.2307/3001968

[43] Luke Zappia, Belinda Phipson, and Alicia Oshlack. 2017. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology* 18, 1 (Sep 2017), 174. https://doi.org/10.1186/s13059-017-1305-0

[44] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. {PrivSyn}: Differentially Private Data Synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*. 929–946.

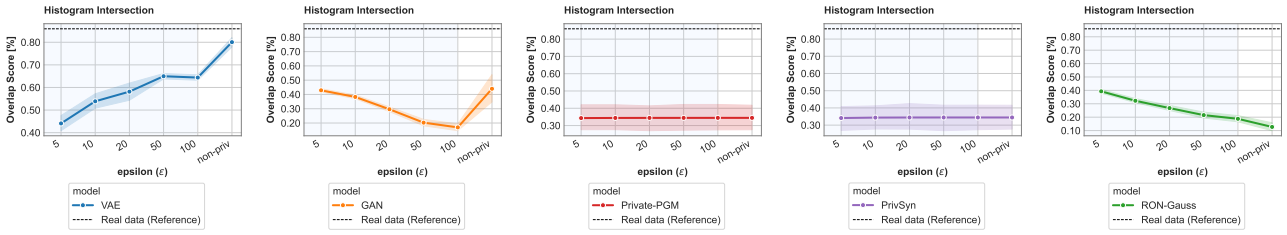
A UTILITY AND STATISTICAL EVALUATION

A.1 Plot of Evaluation Metrics Across Individual Models without Fixing y-axis limit



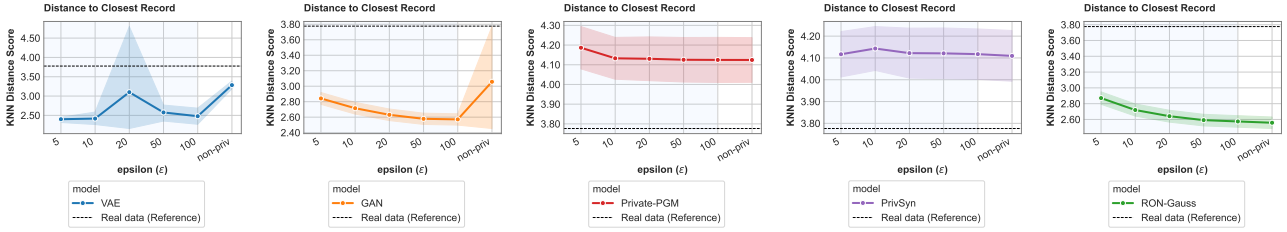
(a) VAE (b) GAN (c) Private-PGM (d) PrivSyn (e) RON-Gauss

Figure 5: Utility Evaluation by Machine Learning Efficacy.



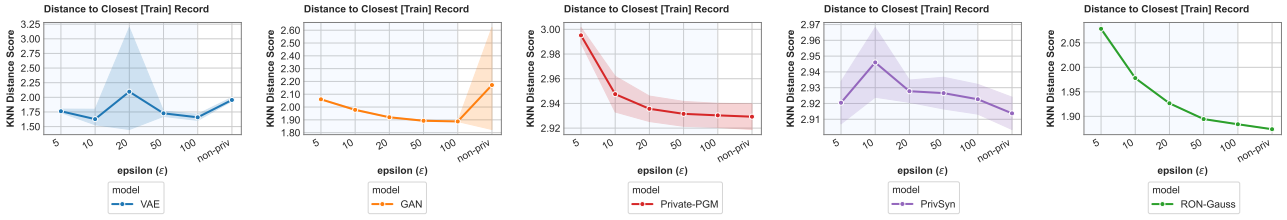
(a) VAE (b) GAN (c) Private-PGM (d) PrivSyn (e) RON-Gauss

Figure 6: Statistical Evaluation by Histogram Intersection.



(a) VAE (b) GAN (c) Private-PGM (d) PrivSyn (e) RON-Gauss

Figure 7: Distance to Closest Record.



(a) VAE (b) GAN (c) Private-PGM (d) PrivSyn (e) RON-Gauss

Figure 8: Distance to Closest (Train) Record.

A.2 Distance to Closest Train Record

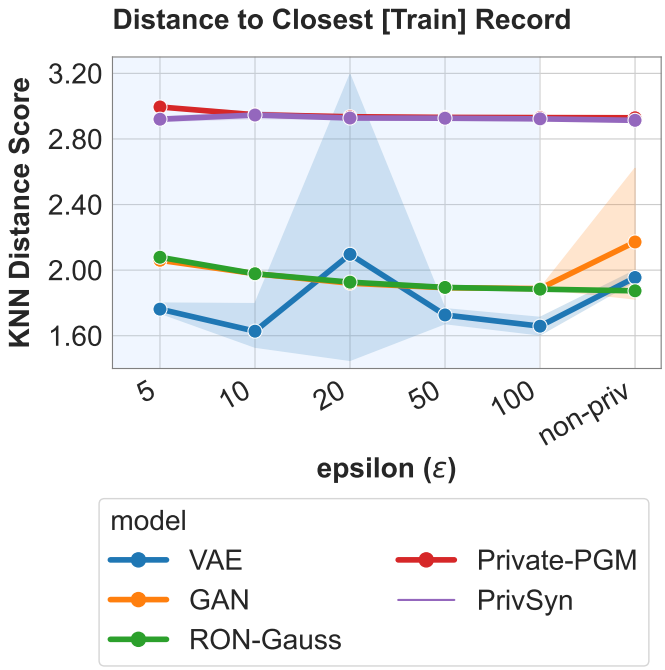


Figure 9: Distance to the Closest Train Record. There is no real data (Reference) since the reference is to the train data. A Score of 0 would signify synthetic data that exactly replicates the training data.

A.3 Correlation between Marginal Metrics

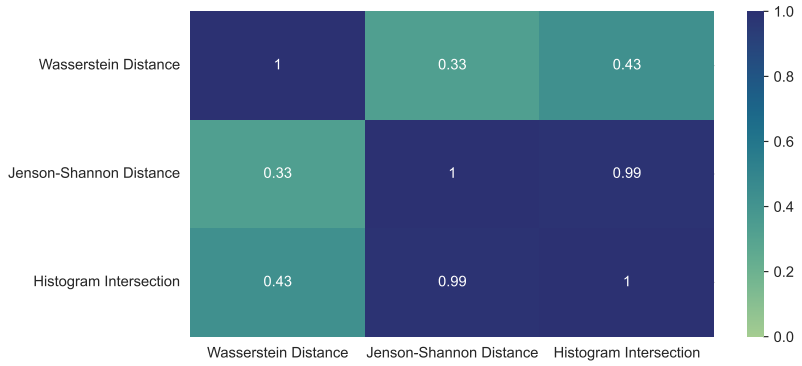
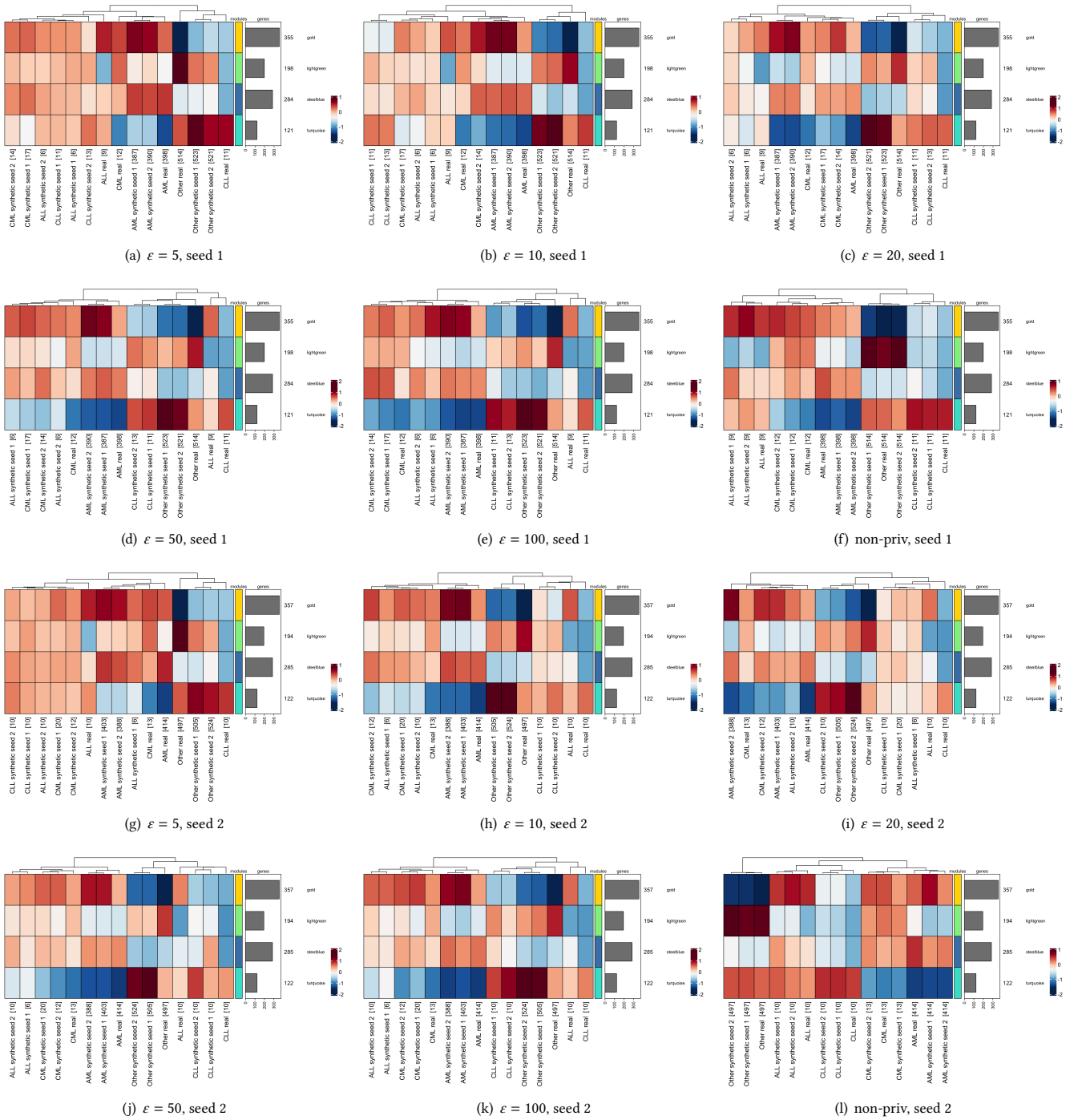


Figure 10: Correlation Coefficients between Marginal Metrics in Absolute Value. A higher score closer to 1 indicates stronger correlation.

B ADDITIONAL PLOTS FOR GENE CO-EXPRESSION EVALUATION

B.1 $r > 0$ (Default Setting)



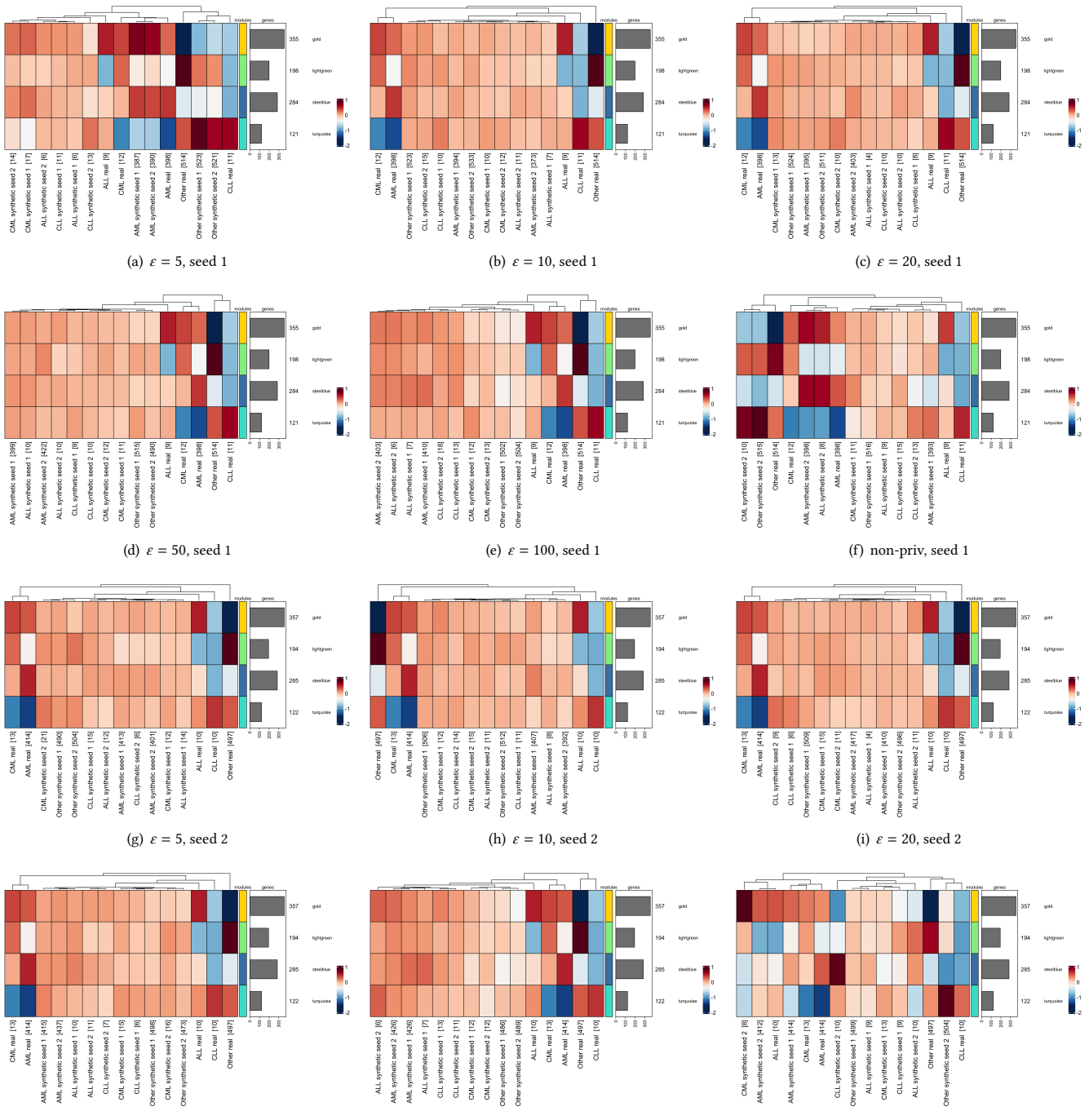
(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

Figure 11: Activation patterns of co-expressed gene modules in VAE after filtering co-expressions for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. While the synthetic data maintains the expression patterns of modules in the non-private setting, they gradually decay with decreasing ε.



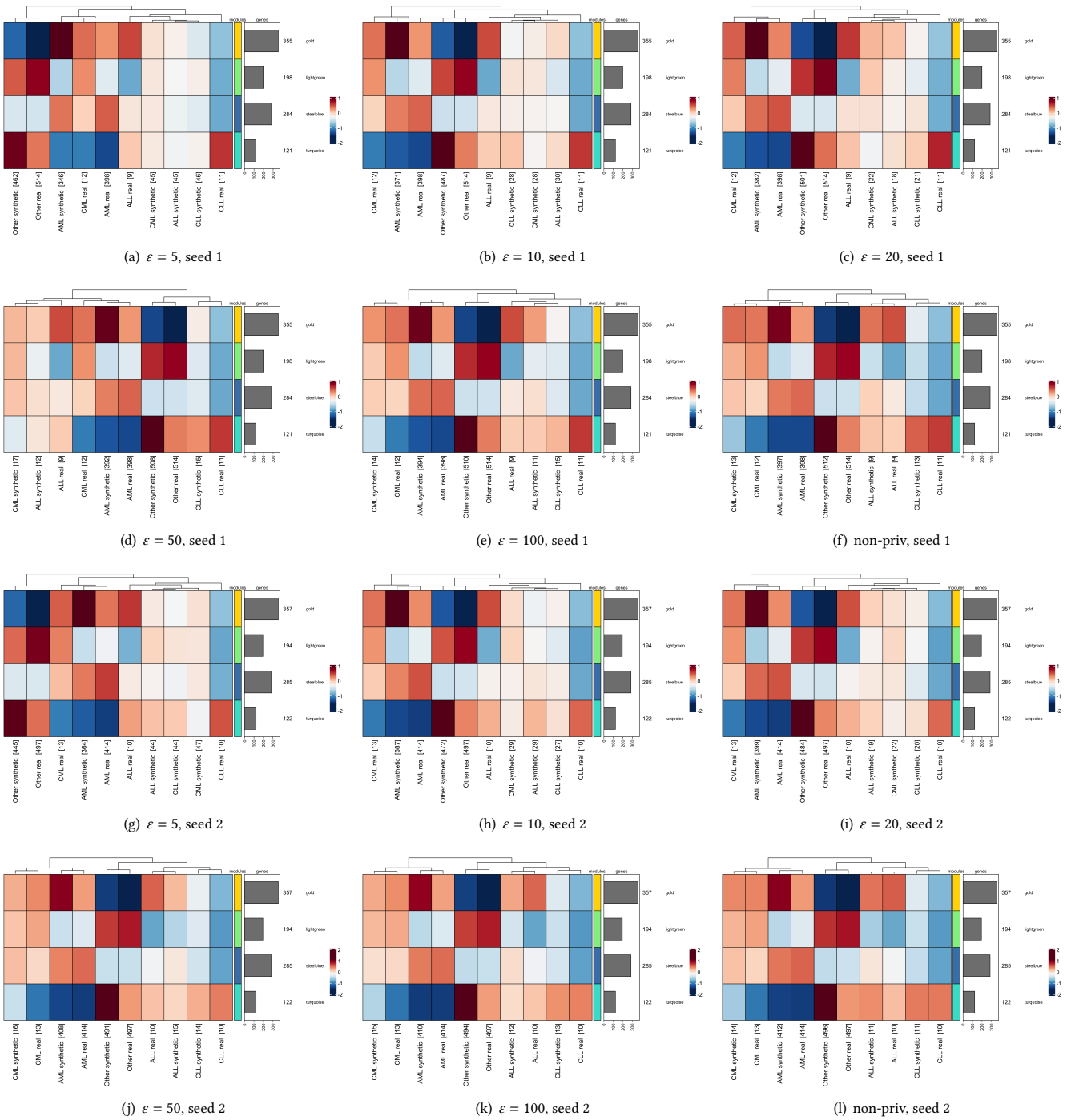
(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

Figure 12: Activation patterns of co-expressed gene modules in GAN after filtering co-expressions for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. The loss of structure in the module activation patterns of the synthetic data is striking, even for high privacy budgets and the non-private setting.



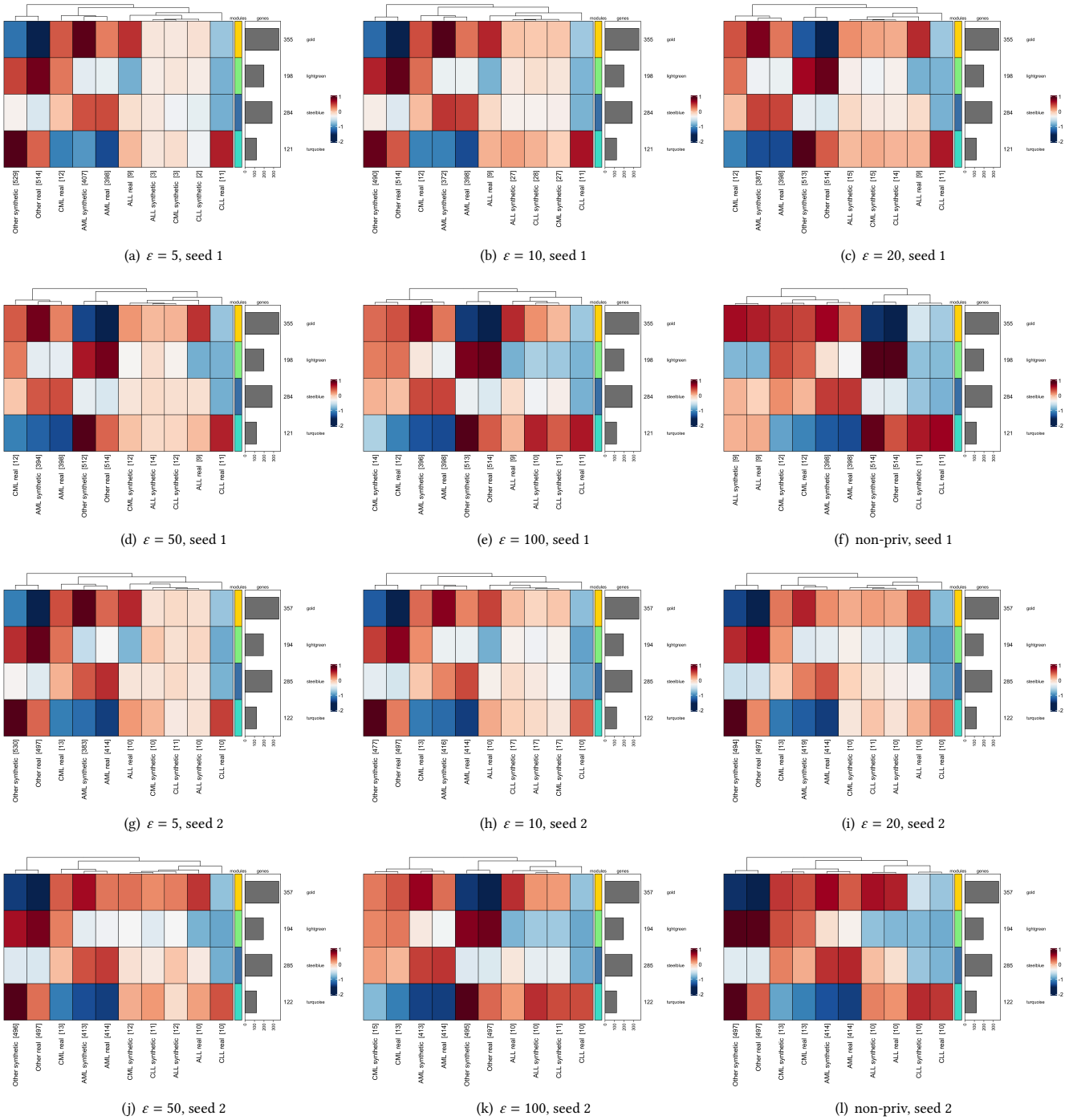
(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

Figure 13: Activation patterns of co-expressed gene modules in PGM after filtering co-expressions for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. The synthetic data exhibits a visible loss of module activation patterns for $\varepsilon \leq 20$.



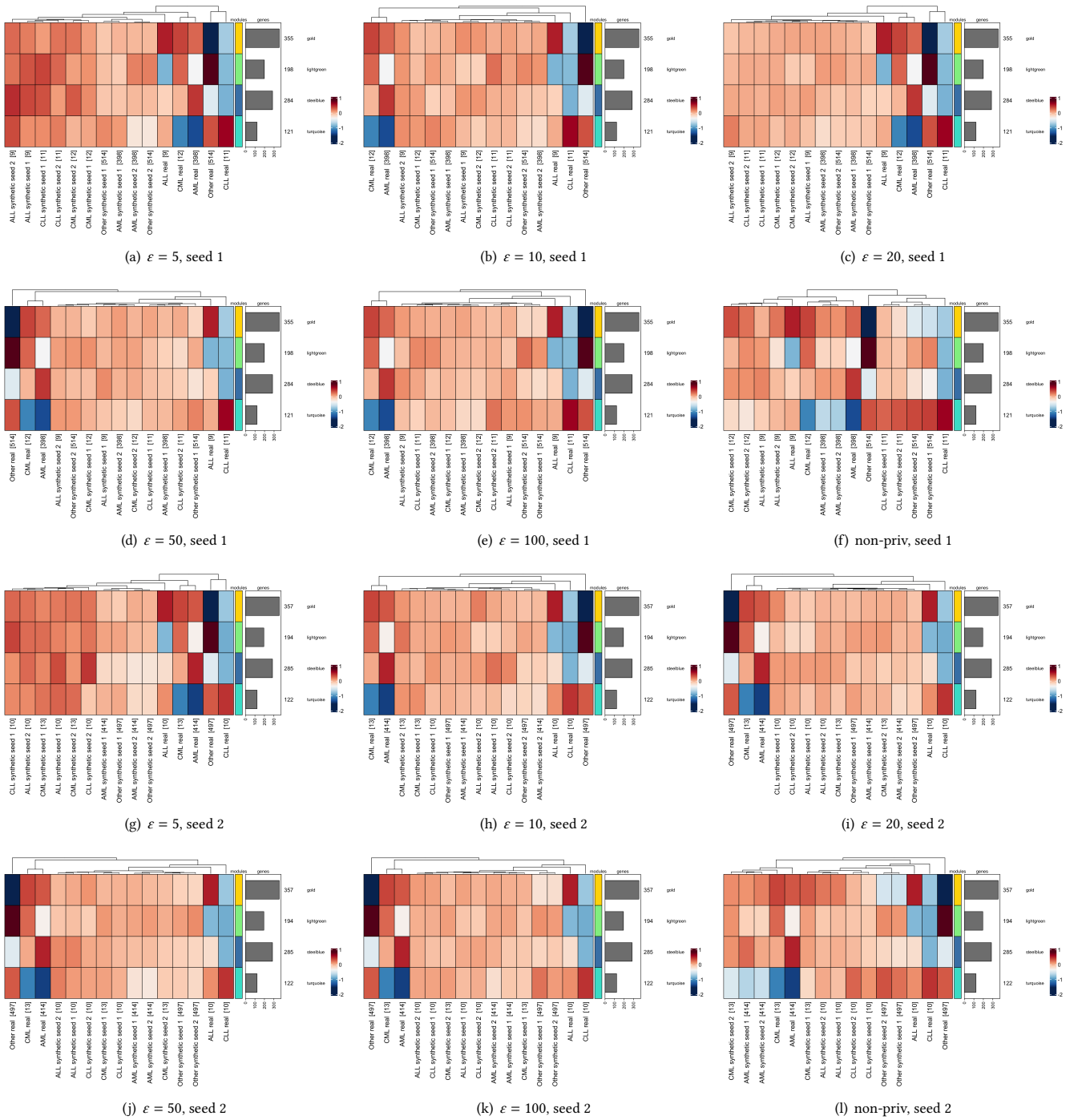
(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

Figure 14: Activation patterns of co-expressed gene modules in PrivSyn after filtering co-expressions for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. A loss of module activation patterns can be observed for all shown privacy budgets, becoming increasingly prominent with decreasing ε.



(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

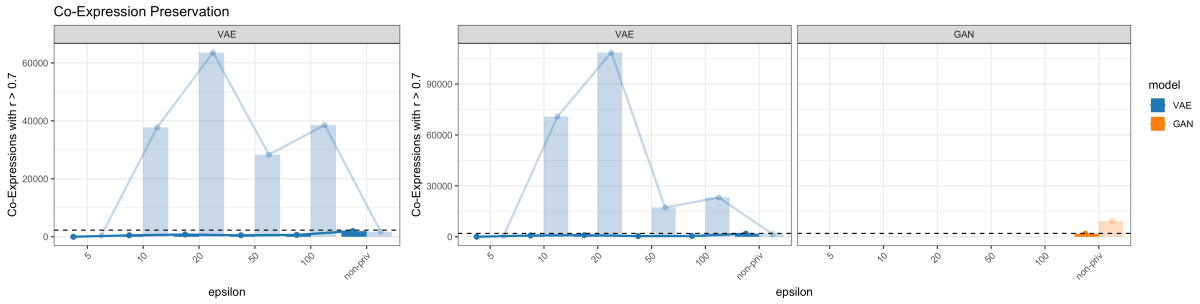
(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

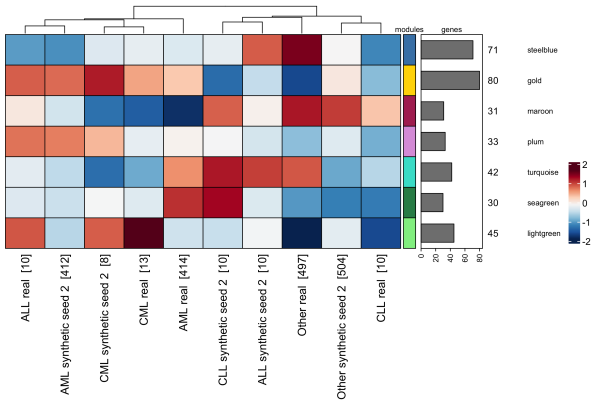
Figure 15: Activation patterns of co-expressed gene modules in RON-Gauss after filtering co-expressions for $r > 0$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. Already in the non-private setting, the synthetic data exhibits mostly homogeneous activation of the different gene modules, maintaining almost none of the structure present in the real data.

B.2 $r > 0.7$



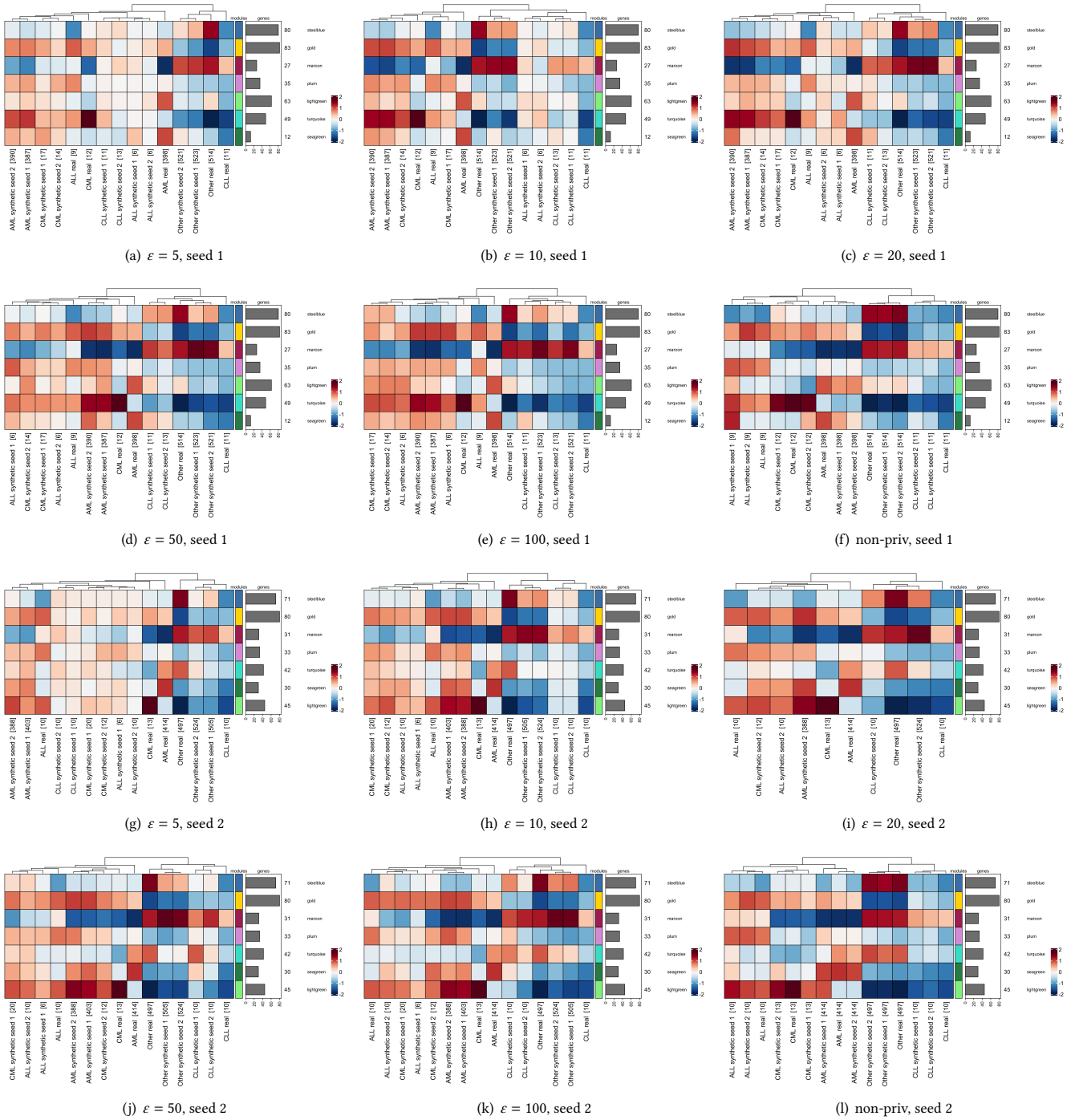
(a) Co-Expression Preservation

Figure 16: Biological Evaluation by Co-Expression Preservation for $r > 0.7$. Shown is the co-expression preservation across the tested models for different values of ε as well as the non-private case for two different seeds used for creating the training split (left and right plot). Note that in the first split (left) only the VAE model was capable of generating significant co-expressions with $r > 0.7$, while in the second seed (right) also the GAN trained without DP yielded some co-expressions. Non-transparent bars give the number of correctly reconstructed co-expressions with Pearson Correlation Coefficient $r > 0.7$ and an associated p-value < 0.05, while semi-transparent bars give the number of co-expressions introduced by the model that did not exist in the real data. The dashed black line indicates the number of co-expressions in the real data. All values shown are means across two different seeds set for generating the data. In the VAE, co-expressions that were falsely introduced in the synthetic data strongly outweigh the correctly reconstructed ones, while the GAN struggles to produce any co-expressions above 0.7, regardless of correct or incorrect.



(a) non-priv, seed 2

Figure 17: Activation patterns of co-expressed gene modules in GAN after filtering co-expressions for $r > 0.7$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. The activation patterns of co-expression modules are poorly maintained in the synthetic data, indicated by the incorrect clustering of disease classes in the real and synthetic data.



(a) $\varepsilon = 5$, seed 1 (b) $\varepsilon = 10$, seed 1 (c) $\varepsilon = 20$, seed 1

(d) $\varepsilon = 50$, seed 1 (e) $\varepsilon = 100$, seed 1 (f) non-priv, seed 1

(g) $\varepsilon = 5$, seed 2 (h) $\varepsilon = 10$, seed 2 (i) $\varepsilon = 20$, seed 2

(j) $\varepsilon = 50$, seed 2 (k) $\varepsilon = 100$, seed 2 (l) non-priv, seed 2

Figure 18: Activation patterns of co-expressed gene modules in VAE after filtering co-expressions for $r > 0.7$. Shown are the Group Fold Changes (GFCs) of gene modules (rows) in the real and the synthetic data sampled with two different seeds. Numbers on the right indicate the number of genes per module. Darker shades of red imply activation of the gene module, while darker shades of blue indicate deactivation. The dendrograms show the hierarchical clustering of the classes in the different data sets. A heatmap is shown for each ε twice, once for each seed used to *split* the training data. Each heatmap further features, in addition to the real data, data from two synthetic sets, one for each seed used to *generate* the data. Note that only one data generation seed yielded any co-expressions above > 0.7 in case of $\varepsilon = 20$, data split seed 2. A general fading of module activation can be observed for decreasing privacy budgets, indicating poor reconstruction of module activation patterns in the synthetic data.