© Indian Academy of Sciences

Review

Leveraging deep learning algorithms for synthetic data generation to design and analyze biological networks

Srisairam Achuthan[1]* , Rishov Chatterjee[1], Sourabh Kotnala[2], Atish Mohanty[2], Supriyo Bhattacharya[3], Ravi Salgia[2] and Prakash Kulkarni[2]*

[1]*Division of Research Informatics, Center for Informatics, City of Hope National Medical Center, Duarte, CA 91010, USA*

[2]*Department of Medical Oncology and Therapeutics and Department of Systems Biology, City of Hope National Medical Center, Duarte, CA 91010, USA*

[3]*Translational Bioinformatics, Department of Computational and Quantitative Medicine, Center for Informatics, City of Hope National Medical Center, Duarte, CA 91010, USA*

Corresponding authors (Emails, sairam.achuthan@gmail.com; pkulkarni@coh.org)

The use of synthetic data is gaining an increasingly prominent role in data and machine learning workflows to build better models and conduct analyses with greater statistical inference. In the domains of healthcare and biomedical research, synthetic data may be seen in structured and unstructured formats. Concomitant with the adoption of synthetic data, a sub-discipline of machine learning known as deep learning has taken the world by storm. At a larger scale, deep learning methods tend to outperform traditional methods in regression and classification tasks. These techniques are also used in generative modeling and are thus prime candidates for generating synthetic data in both structured and unstructured formats. Here, we emphasize the generation of synthetic data in healthcare and biomedical research using deep learning methods for unstructured data formats such as text and images. Deep learning methods leverage the neural network algorithm, and in the context of generative modeling, several neural network architectures can create new synthetic data for a problem at hand including, but not limited to, recurrent neural networks (RNNs), variational autoencoders (VAEs), and generative adversarial networks (GANs). To better understand these methods, we will look at specific case studies such as generating realistic clinical notes of a patient, the generation of synthetic DNA sequences, as well as to enrich experimental data collected during the study of heterotypic cultures of cancer cells.

Keywords. Artificial intelligence; deep learning; generative adversarial networks; machine learning; synthetic data; variational autoencoders

1. Introduction

Imagine a situation in which sufficient and more pertinent data can be made available no matter what scientific hypothesis needs to be tested or theory needs to be validated. Is this even feasible, and if so, would the

data created to meet these lofty goals be acceptable to the scientific community? As strange as all this seems, recent developments in a class of algorithms within the field of artificial intelligence (Yu *et al.* 2018), known as deep learning algorithms (Esteva *et al.* 2019), are making the above seemingly impossible scenario possible in cases where there is some real data readily available. By real data what is meant is that data that have been collected from existing entities (such as

---

This article is part of the Topical Collection: Emergent dynamics of biological networks.

Published online: 12 July 2022

patients in the case of clinical sciences) or those that are generated experimentally in the life sciences or physical sciences to test a specific hypothesis (or set of hypotheses) or check the validity of a proposed theory.

Within basic sciences, different types of biological networks play vital roles in cancer progression, cancer resistance, neuronal rhythms, cardiac rhythms, etc. (Barabási and Oltvai 2004; Koutrouli *et al.* 2020; Muzio *et al.* 2021). Experimentally generated data are never sufficient to study the complexities generated by these biological networks at various temporal and spatial scales. In such cases, an answer to data paucity could lie in the ability to generate synthetic data (Dahmen and Cook 2019; Lindner *et al.* 2019; Goncalves *et al.* 2020; Walonoski *et al.* 2020) that are statistically similar to real data. Statistical similarity implies that the probability distribution of the newly synthetically generated data is similar to that of the original data. Further, the use of synthetic data can lead to the development of scientific solutions that rely on data augmentation, which in this context implies enriching real data with synthetic data. Data augmentation of experimental systems (Hoffmann *et al.* 2019) transforms, in some situations, a data-scarce system into a data-rich system, making them candidates for the application of powerful data analysis techniques like machine learning (Koivu *et al.* 2020; Suh *et al.* 2020; Chen *et al.* 2021) to derive additional insights. Data generated synthetically augments real data points based on their joint probability distribution during the model training process. The way this is achieved is based on the introduction of noise (such as GANs and VAEs, which are discussed below), which in turn leads to a degree of variation that ultimately results in distinct synthetic data points. The inclusion of these unique data points enriches the training set, which in turn leads an algorithm to learn new patterns not originally discovered.

A synthetic data revolution is in progress due to its popularity and prevalence in multiple industries such as self-driving cars, fraud detection, medical imaging as well as healthcare, in general. To accelerate research in the clinical sciences and thereby advance treatment options, it is imperative to overcome the limited access to existing patient data, as well as the difficulties in integrating data from multiple sources within a provider and across providers. To overcome these challenges, virtual cohorts comprising synthetic patients that mirror deeply phenotyped patients are being generated to advance research in diseases such as dementia (Muniz-Terrera *et al.* 2021). Concerns over maintaining patient privacy even with the application of patient de-identification methodologies like safe-harbor and expert determination (*https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/understanding/coveredentities/De-identification/hhs_deid_guidance.pdf*) along with added regulations such as GDPR (*https://www.gartner.com/en/newsroom/press-releases/2020*) are other compelling reasons to move towards the use of synthetic data for secondary research, data commercialization partnerships, educating medical professionals, or software testing purposes. Synthetic datasets have also been found to be useful for validating algorithms created to infer the structure of gene regulatory networks based on expression data (Van den Bulcke *et al.* 2006). Van den Bulcke *et al.* (2006) developed a network generator to create synthetic transcriptional regulatory networks, which in turn was used to simulate gene expression data that approximated experimental data. Other situations where synthetic data can assist with in healthcare include precision medicine studies, modeling simulations, machine learning tasks that involve class imbalance to increase the count of minority samples, as well as rare diseases (*https://www.researchsquare.com/article/rs-116297/v2*). Synthetic data can be created based on real unstructured data such as images, text, audio, or video as well as real structured data as in the case of tabular data (*https://www.statice.ai/post/types-synthetic-data-examples-real-life-examples*). One of the approaches is through generative models (Hazra and Byun 2020; Lan *et al.* 2020), which is a class of deep learning algorithms that can learn from the underlying patterns of real data including statistical properties like probabilistic distributions. Here, we wish to elaborate synthetic data generation methodologies in the context of biomedical case studies including the pursuit to comprehend complex biological networks underpinning cancer resistance. Specifically, we elucidate the use of synthetic data derived from deep learning algorithms to facilitate the formation of a synthetic data layer in a futuristic computational platform designed to study biological networks.

2. Deep learning architectures

2.1 *A primer on RNNs and LSTM networks*

Recurrent neural networks (RNNs) are a specific architecture of neural networks (Pandit and Garg 2021) designed to model any data that are sequential in nature, for example, DNA, audio, financial transactions, etc. They were first introduced by David Rumelhart

(Williams *et al.* 1986). To perform predictive modeling on sequential data, RNNs need the ability to 'remember' past information. The method adopted by this architecture to retain memory is by utilizing its units that resemble the events in sequence. These units are known as RNN unrolled units, and each unit has an input corresponding to the data at a specific time step (e.g., the first unit will have the data input that comes first in the sequence, and the fourth unit will have the data input that comes fourth in the sequence). The parameters of an RNN include the batch size of the input data, the size of the output node, the number of time steps to include in the sequence, and the number of features to include in one time step. This means that the RNN's final output would be three-dimensional (batch size, number of time steps, and output node size) (*https://towardsdatascience.com/all-you-need-to-know-about-rnns-e514f0b00c7c*). Mathematically, memory in an RNN is modeled in the sequential nature of a hidden state vector. Throughout the network, the hidden state also has weights and biases just like the attributed values in an artificial neural network (ANN). Before going through any RNN units, both values are initialized at 0. As the RNN propagates from the input data layer to the hidden layers and then to the output layer and back-propagated, the hidden state at each specific time step is calculated by multiplying the input with its respective weights to which its input bias is added. This calculation then becomes the input to an activation function, typically a hyperbolic tangent. This function is then added to another term which comprises the hidden layer's weight multiplied by the hidden state vector at the previous time step (0 if time step is the first time step) and is then added to the hidden layer's bias. To get the output at a time step, the hidden state vector at the current time step is multiplied by the weight obtained from the output at that time step and added to the output layer's bias obtained at that time step (*https://builtin.com/data-science/recurrent-neural-networks-and-lstm*). When the next hidden state calculation occurs for the following time step, the previous time step's hidden state vector is factored into the computation, allowing the RNN to effectively retain information learned from the network at the previous time step while being able to optimize weights for input at the current time step. This process repeats sequentially until all time steps have a respective output at their corresponding point in time via the sequence. Every time the RNN moves from one unit in time to the next unit of time, we say the network has 'unrolled'. Once the entire network has gone through an entire batch of data, the process will repeat until all batches of data have been trained on the network (figure 1) (*https://towardsdatascience.com/all-you-need-to-know-about-rnns-e514f0b00c7c*). Two problems arise with traditional RNNs when the gradient descent algorithm assigns extremely high values to weights (exploding gradient) or extremely low values to weights (vanishing gradient) to the point where the RNN is no longer able to learn and experiences a tremendous slowdown in its process. To resolve these issues, a new architecture was introduced known as Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997). LSTM networks have a particular cell in the network known as a gated cell. Gated cells determine whether information learned at a particular time step is relevant for the RNN in future time steps by deciding to store or delete information based on the importance of weights at that given time step. The gated cell has three types of gates: input, forget, and output. The information passing through the gated cell goes through a sigmoid function. Since sigmoid functions output a value between 0 and 1, a value of 0 indicates that no information should be sent to the output for the next time step to learn and a value of 1 indicates that all the information at that point in time should be factored into the computation for the next time step's state (*https://medium.com/@humble_bee/rnn-recurrent-neural-networks-lstm-842ba7205bbf*). Values between 0 and 1 indicate that some information should be let through for the next time step to encompass in its hidden state computation and the remainder of information should go through the 'forget gate' to be deleted from the learned memory.

Ultimately, LSTM networks solve the vanishing gradient problem and significantly speed up the learning process of the RNN by still allowing the network to not only learn from the previous time step, but also gain useful information from former time steps as directed by the weights learned and outputs obtained in the gated cells.

But how do we use an LSTM network for generating text? Words are essentially sequences made up of characters. Keeping this idea in mind, the problem can be framed for the LSTM network to predict the next letter in a word. A vocabulary can be generated by creating a list of all unique characters in a document and then each character can be mapped to a unique integer (*https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/*). Using an arbitrary sequence length (n), the document's text can be transformed into subsequences fixed to set up the output as being the last character following $n-1$ characters in a sequence. The LSTM network's goal would then be to
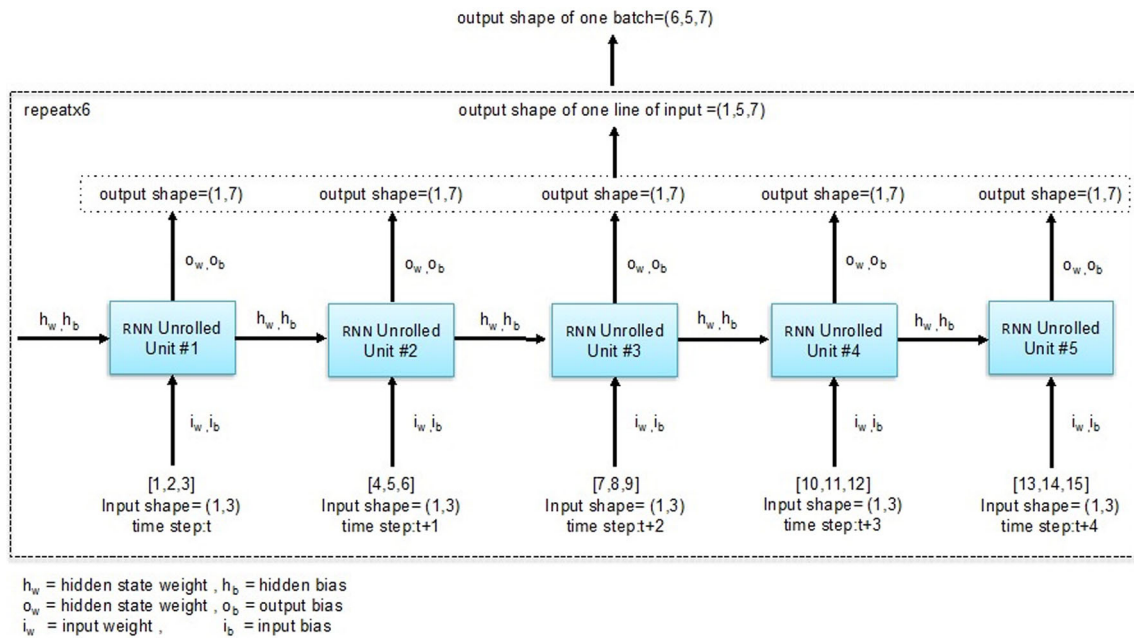
output shape of one batch=(6,5,7)

repeatx6   output shape of one line of input =(1,5,7)

| output shape=(1,7) | output shape=(1,7) | output shape=(1,7) | output shape=(1,7) | output shape=(1,7) |

o_w, o_b o_w, o_b o_w, o_b o_w, o_b o_w, o_b

h_w, h_b → RNN Unrolled Unit #1 → h_w, h_b → RNN Unrolled Unit #2 → h_w, h_b → RNN Unrolled Unit #3 → h_w, h_b → RNN Unrolled Unit #4 → h_w, h_b → RNN Unrolled Unit #5

i_w, i_b i_w, i_b i_w, i_b i_w, i_b i_w, i_b

| [1,2,3] Input shape= (1,3) time step:t | [4,5,6] Input shape= (1,3) time step:t+1 | [7,8,9] Input shape= (1,3) time step:t+2 | [10,11,12] Input shape= (1,3) time step:t+3 | [13,14,15] Input shape= (1,3) time step:t+4 |

h_w = hidden state weight , h_b = hidden bias
o_w = hidden state weight , o_b = output bias
i_w = input weight , i_b = input bias

Figure 1. Schematic of a RNN architecture with 5 time steps. Adapted from *https://towardsdatascience.com/all-you-need-to-know-about-rnns-e514f0b00c7c*.

predict the probability of the next character being one of the unique characters in the vocabulary. The character with the highest probability is chosen as the next character to be added in the sequence. To generate text from the trained LSTM network, a reverse mapping from integers back to the original characters is required. The generation essentially predicts the next character, so one can simply provide a starting sequence and then the LSTM network can predict the next character in the sequence and iteratively continue this process indefinitely to generate new text. LSTM layers in a RNN can also be stacked at the same time step (*https://machinelearningmastery.com/stacked-long-short-term-memory-networks/*). This allows for a deeper representation of memory for the sequence and can be useful for predictions in long and complex sequences.

2.2 A primer on GANs and VAEs

Although LSTM networks and RNNs are powerful models for predicting information in a sequence, they are not necessary for data that do not follow a sequence. One deep learning algorithm that can generate data that is not necessarily in a sequence is known as a generative adversarial network (GAN). GANs are a major advancement in deep learning (Goodfellow *et al.* 2014). The premise behind GANs is to generate complex random variables that follow a specific probability distribution. GANs have shown to perform well when generating images that are unique but resemble realistic properties of a sample of images used during training. However, GANs can be applied to any data that can form a probability distribution. GANs can also be applied to generating realistic data inputs in the form of tabular data (e.g., TabularGAN; Xu and Veeramachaneni 2018), making it a prime candidate for generating synthetic rows in an electronic health record (EHR). Essentially, the architecture comprises two neural networks known as the generator and discriminator. The discriminator network tries to find the boundary that separates real and generated data, while the generator network aims to generate synthetic data that resemble the distribution of the real data (figure 2) (*https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29*). In other words, the generator and discriminator participate in a minimax two-person game (Goodfellow *et al.* 2014). GANs are very fragile architectures which often give rise to difficulties in training them. Due to these challenges, it is quite common that a Nash equilibrium is not attained, and thus more mathematical techniques are
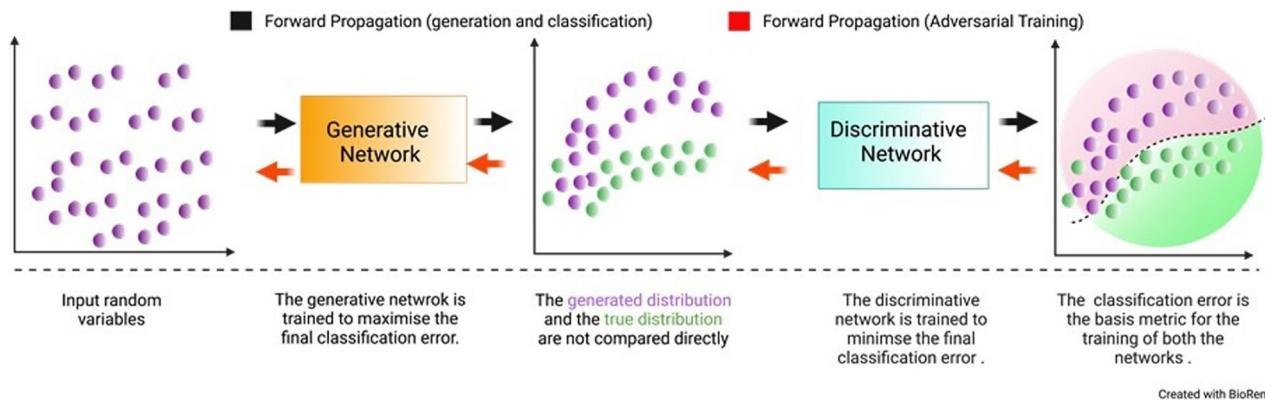
Figure 2. Roles of generative network and discriminative network. Adapted from *https://towardsdatascience.com/ understanding-generative-adversarial-networks-gans-cd6e4651a29*.

required to stabilize the training process and converge approximately. One of these methods is known as proximal training (Farnia and Ozdaglar 2020).

Mathematically, the generator network is optimized to produce data samples in future iterations by minimizing the distance between the real and synthetic data distributions such that the discriminator network faces challenges in discriminating synthetic data from real data. Essentially the generator network starts off by generating inputs in a uniform, random distribution and compares its values with a probability distribution of interest (e.g., a high-resolution image of a melanoma tumor cell, or patient profiles in an electronic medical record). The comparison is done through the calculation of a distance known as maximum mean discrepancy (MMD), which estimates the distance between two probability distributions from provided samples. Within a GAN, back-propagation minimizes the MMD through the perspective of the generator network by updating the weights to increase the discriminator's classification error while the discriminator network's weights are updated to minimize classification error. This optimization design results in an enriched dataset which also draws from modeling the noise around the distribution that the real data follows.

The ideal scenario in this approach is that the training stops when the discriminator network predicts a probability of 0.5 for either the true or synthetic data distributions as that would imply that the data generated by the generator network resembles the real data distribution perfectly. There are also additions to GANs that allow them to be used for generating data in the form of sequences. GANs also suffer from various limitations including overfitting, mode collapse, and vanishing gradient. To address these concerns, techniques to improve GANs are under development.

2.3 *Overfitting*

The GAN generates synthetic data based on the probability distribution of the original data during training but fails to generalize during testing when new data represent a significantly different probability distribution. Solutions to prevent overfitting while training a GAN have been presented such as momentum, dropout, and regularization (Arjovsky and Bottou 2017; Roth *et al.* 2017; Lee and Seok 2020).

2.4 *Mode collapse and vanishing gradient*

Generator collapses cause a limited variety of newly generated synthetic samples that are distinct from one another after the GAN has been trained. Adding distinct real data samples is how we can resolve this issue. Since the collapse causes a vanishing gradient, we can tune the learning rate of the training design to avoid the Cold Start Problem (*https://towardsdatascience.com/ the-cold-start-problem-with-artificial-intelligence-4993 8ed3f612*).

As seen above, GANs are very flexible architectures for generating synthetic data no matter how complex the input data is. Although GANs generate high-quality data, they do take a long time to train. Another technique for generating synthetic data is known as a variational autoencoder (VAE). VAEs are commonly used for creating higher-quality images out of low-quality ones (image denoising) or generating images that are like those used for training. Essentially, a VAE can learn the underlying probability distribution of the data it is trained on. New data can then be sampled from the learned distribution. The mechanism of learning the probability distribution of the data

during training is known as latent variable representation (Doersch 2016).

The goal with VAEs is to find the posterior distribution $p(z|x)$, which is the distribution of the encoded variable given the decoded variable. The prior distribution of the encoder's probability, $p(z)$, is assumed to be a standard Gaussian distribution. The distribution of the decoded variable given the encoded one is $p(x|z)$, which is a Gaussian distribution with a mean that represents a deterministic function of the z variable. Its covariance matrix has a positive constant c that can be multiplied with the identity matrix I. Using Bayes' theorem, it is possible to get an equation for finding $p(z|x)$, but there is no closed-form solution, and hence variational inference is used for approximation. In VAEs, this approximation is done by finding a Gaussian distribution $Q(z)$ for which its mean and covariance are defined by the functions $g(x)$ and $h(x)$. Finding the best approximation for $p(z|x)$ occurs by minimizing the Kullback–Leibler (KL) divergence between the approximation $Q(z)$ and $p(z|x)$. Since the functions f, g, and h are not known, a neural network known as the encoder network is utilized to approximate $g(x)$ and $h(x)$. Another neural network known as a decoder network is used to approximate $f(z)$. These two networks are concatenated by the latent layer which acts as a regularization term through the KL divergence in the network to avoid overfitting. The loss function of the VAE is the negative log-likelihood (Doersch 2016) with the KL divergence as the regularizer.

2.5   Applications of deep learning architectures to generate synthetic data in biological networks

Case study 1: Generating variable-length DNA sequences that are likely to code for antimicrobial peptides

Antimicrobial peptides (AMPs) play a critical role in addressing the looming global health crisis of antibiotic resistance exhibited by different types of pathogens. Discovering new DNA sequences that could likely translate to AMPs, experimentally, is both costly and challenging for researchers to implement in a lab setting. Therefore, the Feedback GAN (FBGAN) architecture (Gupta and Zou 2018) was developed by Stanford researchers to generate gene sequences that encode for variable-length proteins. The architecture utilizes a common variant to the traditional GAN by minimizing a different loss known as Wasserstein divergence (Wu *et al.* 2018) instead of the MMD. It

has been found that Wasserstein divergence helps in making the training of a GAN more stable when varying hyperparameter configurations. The discriminator network is a convolutional neural network (CNN) with five residual layers that contain two 1D convolutions. The output layer in the discriminator is a Gumbel–Softmax layer (Jang *et al.* 2016) instead of a Softmax layer. In the generator network, the argmax of the probability distribution is taken to output a single nucleotide at each position. The way feedback is introduced to this GAN is through a component known as an analyzer which is an RNN that can take in a gene sequence and predict the probability that the sequence will code for an AMP. If the probability of the DNA sequence coding for an AMP is greater than 0.8, the sequence goes back to the discriminator network and is classified as a real sequence. This mechanism allows for the generator network to generate DNA sequences that are more likely to code for an AMP over time, which means that the generated DNA sequences optimize the protein function once encoded (figure 3) (Gupta and Zou 2018).

Case study 2: Generating synthetic electronic health records (EHR) to mimic a scalable repository of structured and unstructured patient data

For educational purposes, it is important to generate synthetic EHR data to train the next generation of physicians based on realistic extracts of observational healthcare records of patients (*https://www.researchsquare. com/article/rs-116297/v2*). The structured data are tabular in nature which captures information as rows and columns. Xu *et al.* (2019) found that traditional GANs performed poorly, in comparison with baseline methods, to model the probability distribution of rows in tabular data especially regarding metrics such as likelihood fitness and machine learning efficacy of the synthetically generated data. Tabular data can be challenging to model since they typically contains a mix of continuous and discrete columns. Tabular GANs (TGANs; Xu and Veeramachaneni 2018) are capable of synthesizing both continuous and discrete columns after a pre-processing step is applied to normalize categorical and discrete columns using Gaussian Mixture Models (GMMs). However, continuous columns may have multiple modes of non-Gaussian values and discrete columns that may be hampered by severe imbalance. To address these, Xu *et al.* (2019) proposed conditional tabular GANs (CTGANs) wherein a mode-specific normalization was introduced to overcome the non-Gaussian and multimodal distributions along with the Wasserstein loss for greater stability. The mode-specific normalization converts continuous values of
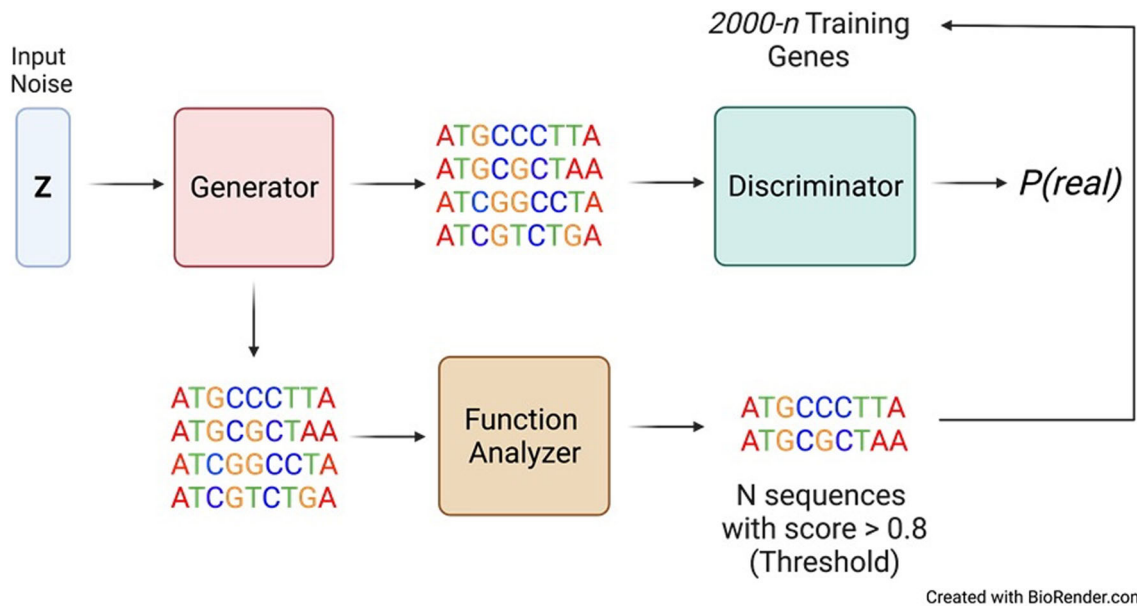
Figure 3. Schematic of the FBGAN Architecture. Adapted from Gupta and Zou (2018).

arbitrary range and distribution into a bounded vector representation suitable for neural networks. Apart from this, a conditional generator was designed along with training-by-sampling to overcome the imbalanced training data issue that is likely to arise from columns with discrete values.

There is also an unstructured component to EHRs which are notes written by physicians for their patients. Similar to the generation of tabular data, there is a lot of utility in generating unstructured medical documents such as progress notes and pathology reports for educational purposes. However, these documents are often scarce due to regulatory issues surrounding the sharing of these documents across stakeholders. Generating synthetic medical documents implies that new samples of data do not belong to a real patient and hence any privacy concerns are non-existent. Researchers at IBM built a 2-layer LSTM network with 650 hidden units trained on processed data sources: MedText-2 and MedText-103, which are cleaned from the clinical notes in the MIMIC-III dataset (Johnson *et al.* 2016). This model was designed to predict the next word in a sentence and thus the vocabulary consisted of unique words instead of unique characters. The researchers then generated new words from the trained LSTM network to have the same word count as the original MedText-2 and MedText-103 clinical note datasets. Despite receiving validation from clinicians, the generated notes do not always make clinical sense for a patient and there is also a tendency for poor grammar to occur. Other issues included switching of the patient's gender and short-term text generation on diseases that

do not make sense, such as 'hepatitis C deficiency'. Although there are challenges to address, the synthetic notes do also contain properties that resemble the clinical notes in MIMIC-III. This has the potential to impact medical education.

Case study 3: Leveraging deep learning algorithms to study heterotypic cultures of cancer cells

Resistance to chemotherapy is a major impediment in treating cancer. Resistance is generally held to primarily arise through random genetic mutations and the subsequent expansion of mutant clones via Darwinian selection (Greaves and Maley 2012; Álvarez-Arenas *et al.* 2019). Hence, the phenomenon has been approached from a fully reductionist, gene-centric perspective (Vogelstein *et al.* 2013). However, it is now evident that drug resistance need not occur through mutations acting alone. Several non-genetic mechanisms including epigenetic modifications and protein interaction network rewiring that leads to phenotypic switching can also impact a cancer cell's ability to develop drug resistance (Huang and Ingber 2006-2007; Sharma *et al.* 2010; Dawson and Kouzarides 2012; Kulkarni *et al.* 2013; Mahmoudabadi *et al.* 2013; Pisco *et al.* 2013; Jones *et al.* 2016; Salgia and Kulkarni 2018; Bell and Gillan 2020). Group behavior emerging from such non-genetic mechanisms can sustain a heterogeneous cancer cell population with multiple interchangeable phenotypes, producing temporary drug tolerance and facilitating the initiation and progression to permanent drug resistance (Wu *et al.* 2014; Zhang *et al.* 2017; Kaznatcheev *et al.*

2019; Stankova 2019; Stankova *et al.* 2019; Bhattacharya *et al.* 2021). One major reason why current cancer therapies fail to prevent drug resistance or tumor recurrence is the lack of in-depth understanding of how cancer cells react to cytotoxic drugs over a prolonged period. The cancer evolutionary landscape is highly complex and dynamically evolving due to the cellular heterogeneity that results from the phenotypic plasticity of the tumor cells. Although there has been significant advancement in the understanding of cancer evolutionary biology and associated theoretical methods, there has been no systematic effort to integrate these methods into a computational platform to explore tumor behavior.

Limited experimental data are major impediments to innovation. However, this limitation can be overcome to some extent by using deep learning methods such as GANs and VAEs, which are capable of generating synthetic heterotypic cultures extending *in vitro* experimental data (Nam *et al.* 2021), following the distribution of a sample taken from real experimental data (figure 4). Potential limitations of synthetic data may include, not accounting for outliers in the original data, susceptibility to statistical noise as well as the need for continuous verification of the augmented data versus that generated experimentally. Testing augmented data appropriately will be necessary. A reinforcement learning paradigm (Sutton and Barto 1998) is dictated by the ability of an agent or group of agents to find policies (rules) in complex environments to maximize reward. The reward in the context of cancer cell modeling is the ability to survive as captured by an increase in the intrinsic growth rate of the community. Agent-based (Marée *et al.* 2007) and evolutionary game theory (EGT) models (Nam *et al.* 2020) simulate cellular behavior based on predefined rules, whereas the reinforcement learning algorithms can discover rules that lead to complex cellular interactions (Hou *et al.* 2019). These methods are still being developed and improved upon by the scientific community, and applying them to cancer evolution will be highly innovative and novel. We recognize that it cannot be predicted *a priori* which methods will work successfully using the available cellular data. However, we are confident that a computational platform will allow us to deploy and test these methods in a rapid and systematic way. These newly discovered algorithms can be used within the traditional discrete or continuum models to further explore effects arising from varying nutrients, stress, and cellular composition. For example, the Deep-Q-Network algorithm (Hou *et al.* 2019) in combination with age-based models simulated cellular motion while incorporating 3D image data of the cellular environment based on multiple rules in the context of reinforcement learning.

3. Challenges in biomedical use of synthetic data generated by GANs and VAEs

Enriching real biomedical datasets with synthetic data generated by GANs and VAEs comes with a trade-off between benefits and challenges. We have discussed the benefits above as they pertain to generating variable-length DNA sequences that could code for AMPs, generating synthetic EHR records with structured and unstructured data as well as for the study of heterotypic cultures of cancer cells. One of the challenges that arise in the use of GANs is the lack of variety in the synthetic data generated due to mode collapse where the generator limits producing new data samples. Other challenges in the use of GANs include
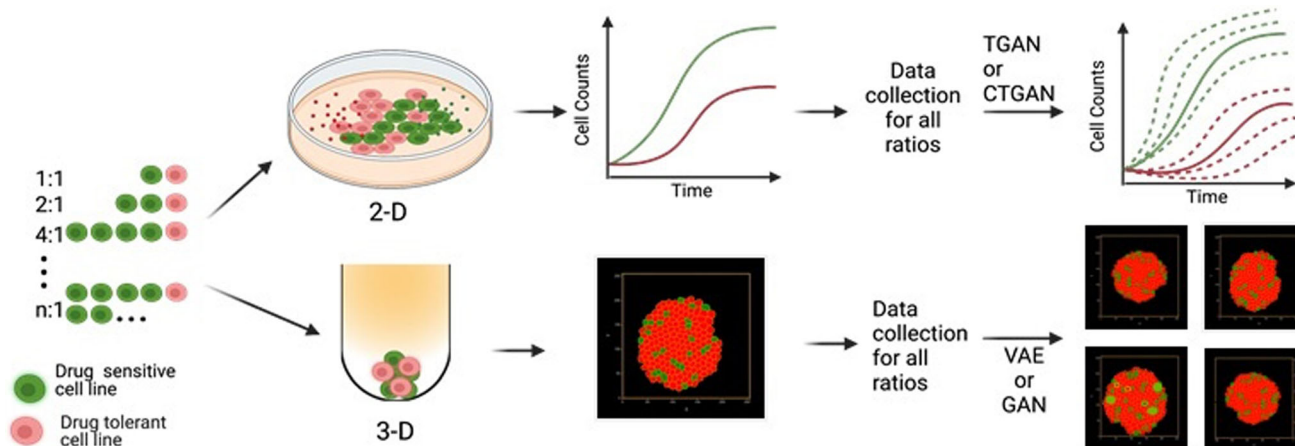

Figure 4. Schematic of synthetic tabular and image data generated using deep learning algorithms (TGAN/CTGAN, VAE/GAN) based on real experimental data from various heterotypic cultures of cancer cells.

sensitivity to neural network architectures and in the selection of hyperparameters. For example, TGANs that are designed to generate synthetic tabular data could suffer from mode collapse. As an active area of research, several GAN architectures such as CTGAN (Xu *et al.* 2019), WGAN (Arjovsky *et al.* 2017), WGAN-GP (Gulrajani *et al.* 2017), DCGAN (Radford *et al.* 2015), etc., have been developed to overcome these challenges. OVAE (Vardhan and Kok 2020), another recently proposed approach based on VAEs, was designed to preserve the distributional characteristics of the real data in its generated data. A novel form of VAE, known as robust VAE (Akrami *et al.* 2020), was proposed to handle tabular datasets with categorical and continuous features that are robust to outliers in the training data. Synthetic tabular data generated by various types of GANs can be evaluated using visual, similarity, statistical, and machine learning-based metrics. Recently, OCT-GANs (Kim *et al.* 2021), an extension of CTGANs, was proposed to process raw tabular data, in the context of web-based research, with a mode-based normalization technique. In the context of image data, VEEGAN (Srivastava *et al.* 2017) was shown to resist mode collapse to a large extent and produce realistic images. AdaGAN (Tolstikhin et al. 2017; Lala et al. 2018) was shown to have superior performance while evaluating mode collapse among selected GAN architectures. Overall, deep learning methods such as GANs and VAEs are continuously being improved to handle different types of data that include, among others, tabular and image data, which are widely used within biomedical applications. In specific contexts, as discussed in this work, the performance of deep learning algorithms will dictate how well the synthetic data generated by them is able to address the data paucity problem that exists currently in analyzing biological networks.

Acknowledgements

Declarations

Conflict of interest The authors declare they have no conflict of interest.

References

Akrami H, Aydore S, Leahy RM and Joshi AA 2020 Robust variational autoencoder for tabular data with beta divergence. *arXiv https://doi.org/10.48550/arXiv.2006.08204*

Arjovsky M and Bottou L 2017 Towards principled methods for training generative adversarial networks. arXiv *https://doi.org/10.48550/arXiv.1701.04862*

Arjovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks. *Proc. 34th Int. Conf. Machine Learn.* 214–223

Álvarez-Arenas A, Podolski-Renic A, Belmonte-Beitia J, Pesic M and Calvo GF 2019 Interplay of Darwinian selection, Lamarckian induction and microvesicle transfer on drug resistance in cancer. *Sci. Rep.* **9** 9332

Barabási AL and Oltvai ZN 2004 Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **5** 101–113

Bell CC and Gilan O 2020 Principles and mechanisms of non-genetic resistance in cancer. *Br. J. Cancer* **122** 465–472

Bhattacharya S, Mohanty A, Achuthan S, *et al*. 2021 Group behavior and emergence of cancer drug resistance. *Trends Cancer* **7** 323–334

Chen RJ, Lu MY, Chen TY, Williamson DFK and Mahmood F 2021 Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **5** 493–497

Dahmen J and Cook D 2019 SynSys: A synthetic data generation system for healthcare. *Sensors* **19** 1181

Dawson MA and Kouzarides T 2012 Cancer epigenetics: from mechanism to therapy. *Cell* **150** 12–27

Doersch C 2016 Tutorial on variational autoencoders. *arXiv https://doi.org/10.48550/arXiv.1606.05908*

Esteva A, Robicquet A, Ramsundar B, *et al*. 2019 A guide to deep learning in healthcare. *Nat. Med.* **25** 24–29

Farnia F and Ozdaglar A 2020 GANs may have no Nash equilibria. *arXiv https://doi.org/10.48550/arXiv.2002.09124*

Goncalves A, Ray P, Soper B, *et al*. 2020 Generation and evaluation of synthetic patient data. *BMC Med. Res. Methodol.* **20** 108

Goodfellow IJ, Pouget-Abadie J, Mirza M, *et al.* 2014 Generative adversarial networks. *arXiv https://doi.org/10.48550/arXiv.1406.2661*

Greaves M and Maley CC 2012 Clonal evolution in cancer. *Nature* **481** 306–313

Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V and Courville A 2017 Improved training of Wasserstein GANs. *arXiv https://doi.org/10.48550/arXiv.1704.00028*

Gupta A and Zou J 2018 Feedback GAN (FBGAN) for DNA: A novel feedback-loop architecture for optimizing protein functions. *arXiv https://doi.org/10.48550/arXiv.1804.01694*

Hazra D and Byun Y-C 2020 SynSigGAN: Generative adversarial networks for synthetic biomedical signal generation. *Biology* **9** 441

Hochreiter S and Schmidhuber J 1997 Long short-term memory. *Neural Comput.* **9** 1735–1780

Hoffmann J, Bar-Sinai Y, Lee LM, *et al*. 2019 Machine learning in a data-limited regime: Augmenting

experiments with synthetic data uncovers order in crumpled sheets. *Sci. Adv.* **5** eaau679

Hou H, Gan T, Yang Y, *et al.* 2019 Using deep reinforcement learning to speed up collective cell migration. *BMC Bioinform.* **20** 571

Huang S and Ingber DE 2006–2007 A non-genetic basis for cancer progression and metastasis: self-organizing attractors in cell regulatory networks. *Breast Dis.* **26** 27–54

Jang E, Gu S and Poole B 2016 Categorical reparametrization with Gumbel-Softmax. *arXiv* https://doi.org/10.48550/arXiv.1611.01144

Johnson AEW, Pollard TJ, Shen L, *et al.* 2016 MIMIC-III, a freely accessible critical care database. *Sci. Data* **3** 160035

Jones PA, Issa JP and Baylin S 2016 Targeting the cancer epigenome for therapy. *Nat. Rev. Genet.* **17** 630–641

Kim J, Jeon J, Lee J, Hyeong J and Park N 2021 OCT-GAN: Neural ODE-based conditional tabular GANs. *arXiv* https://doi.org/10.48550/arXiv.2105.14969

Koivu A, Sairanen M, Airola A and Pahikkala T 2020 Synthetic minority oversampling of vital statistics data with generative adversarial networks. *J. Am. Med. Inform. Ass.* **27** 1667–1674

Kaznatcheev A, Peacock J, Basanta D, Marusyk A and Scott JG 2019 Fibroblasts and alectinib switch the evolutionary games played by non-small cell lung cancer. *Nat. Ecol. Evol.* **3** 450–456

Koutrouli M, Karatzas E, Paez-Espino D and Pavlopoulos GA 2020 A guide to conquer the biological network era using graph theory. *Front. Bioeng. Biotechnol.* **8** 34

Kulkarni P, Shiraishi T and Kulkarni RV 2013 Cancer: Tilting at windmills? *Mol. Cancer* **12** 108

Lala S, Shady M, Belyaeva A and Liu M 2018 Evaluation of mode collapse in generative adversarial networks. *Proceedings of 2018 IEEE High Performance Extreme Computing Conference*

Lan L, You L, Zhang Z, *et al.* 2020 Generative adversarial networks and its applications in biomedical informatics. *Front. Public Health* **8** 16

Lee M and Seok J 2020 Regularization methods for generative adversarial networks: an overview of recent studies. *arXiv* https://doi.org/10.48550/arXiv.2005.09165

Lindner L, Narnhofer D, Weber M, *et al.* 2019 Using synthetic training data for deep learning-based GBM segmentation. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* https://doi.org/10.1109/EMBC.2019.8856297

Mahmoudabadi G, Rajagopalan K, Getzenberg RH, *et al.* 2013 Intrinsically disordered proteins and conformational noise: implications in cancer. *Cell Cycle* **12** 26–31

Marée AFM, Grieneisen VA and Hogeweg P 2007 The cellular potts model and biophysical properties of cells, tissues and morphogenesis; in *Single-Cell-Based Models in Biology and Medicine. Mathematics and Biosciences in Interaction* (eds) ARA Anderson, MAJ Chaplain and KA Rejniak (Basel: Birkhäuser)

Muniz-Terrera G, Mendelevitch O, Barnes R and Lesh MD 2021 Virtual cohorts and synthetic data in dementia: an illustration of their potential to advance research. *Front. Artif. Intell.* **4** 613956

Muzio G, O'Bray L and Borgwardt K 2021 Biological network analysis with deep learning. *Brief Bioinform.* **22** 1515–1530

Nam A, Mohanty A, Bhattacharya S, *et al.* 2020 Suppressing chemoresistance in lung cancer via dynamic phenotypic switching and intermittent therapy. *bioRxiv* doi: https://doi.org/10.1101/2020.04.06.028472

Nam A, Mohanty A, Bhattacharya S, *et al.* Suppressing chemoresistance in lung cancer via dynamic phenotypic switching and intermittent therapy. *bioRxiv* doi: https://doi.org/10.1101/2020.04.06.028472

Pandit A and Garg A 2021 Artificial neural networks in healthcare: a systematic review. *11th International Conference on Cloud Computing, Data Science and Engineering* doi: https://doi.org/10.1109/Confluence51648.2021.9377086

Pisco AO, Brock A, Zhou J, *et al.* 2013 Non-Darwinian dynamics in therapy-induced cancer drug resistance. *Nat. Commun.* **4** 2467

Radford A, Metz L and Chintala S 2015 Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* https://doi.org/10.48550/arXiv.1511.06434

Roth K, Lucchi A, Nowozin S and Hofmann T 2017 Stabilizing training generative adversarial networks through regularization. *arXiv* https://doi.org/10.48550/arXiv.1705.09367

Salgia R and Kulkarni P 2018 The genetic/non-genetic duality of drug "resistance" in cancer. *Trends Cancer* **4** 110–118

Sharma SV, Lee DY, Li B, *et al.* 2010 A chromatin-mediated reversible drug-tolerant state in cancer cell subpopulations. *Cell* **141** 69–80

Srivastava A, Valkov L, Russell C, Gutmann MU and Sutton C 2017 VEEGAN: Reducing mode collapse in GANs using implicit variational learning. *arXiv* https://doi.org/10.48550/arXiv.1705.07761

Staňková K 2019 Resistance games. *Nat. Ecol. Evol.* **3** 336–337

Stankova K, Brown JS, Dalton WS and Gatenby RA 2019 Optimizing cancer treatment using game theory: a review. *JAMA Oncol.* **5** 96–103

Suh S, Lee H, Lukowicz P and Lee OL 2020 CEGAN: Classification enhancement generative adversarial networks for unraveling data imbalance problems. *Neural Netw.* **133** 69–86

Sutton RS and Barto AG 1998 Reinforcement learning: an introduction; in *Adaptive Computation and Machine Learning Series* (eds) RS Sutton and AG Barto (Cambridge: Bradford Book, MIT Press)

Tolstikhin I, Gelly S, Bousquet O, Simon-Gabriel CJ and Schölkopf B 2017 AdaGAN: Boosting generative models. *arXiv* https://doi.org/10.48550/arXiv.1701.02386

Van den Bulcke T, Van Leemput K, Naudts B, *et al*. 2006 SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinform.* **7** 43

Vardhan LVH and Kok S 2020 Synthetic tabular data generation with oblivious variational autoencoders: alleviating the paucity of personal tabular data for open research. *Proceedings of the 37th International conference on machine learning, ICML HSYS Workshop 2020*

Vogelstein B, Papadopoulos N, Velculescu VE, *et al*. 2013 Cancer genome landscapes. *Science* **339** 1546–1558

Walonoski J, Klaus S, Granger E, *et al*. 2020 Synthea™ Novel coronavirus (COVID-19) model and synthetic data set. *Intell. Based Med.* **1** 100007

Williams RJ, Hinton GE and Rumelhart DE 1986 Learning representations by back-propogating errors. *Nature* **323** 533–536

Wu A, Liao D, Tlsty TD, Sturm JC and Austin RH 2014 Game theory in the death galaxy: interaction of cancer and stromal cells in tumour microenvironment. *Interface Focus* **4** 20140028

Wu J, Huang Z, Thoma J, Acharya D and Gool LV 2018 Wasserstein divergence for GANs. *arXiv* https://doi.org/10.48550/arXiv.1712.01026

Xu L and Veeramachaneni K 2018 Synthesizing tabular data using generative adversarial networks. *arXiv* https://doi.org/10.48550/arXiv.1811.11264

Xu L, Skoularidou M, Cuesta-Infante A and Veeramachaneni K 2019 Modeling tabular data using conditional GAN. *arXiv* https://doi.org/10.48550/arXiv.1907.00503

Yu KH, Beam AL and Kohane IS 2018 Artificial Intelligence in healthcare. *Nat. Biomed. Eng.* **2** 719–731

Zhang J, Cunningham JJ, Brown JS and Gatenby RA 2017 Integrating evolutionary dynamics into treatment of metastatic castrate-resistant prostate cancer. *Nat. Commun.* **8** 1816

Corresponding editor: Susmita Roy