Machine Learning: Science and Technology

PAPER • **OPEN ACCESS**

ICEGAN: inverse covariance estimating generative adversarial network

To cite this article: Insoo Kim *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 025008

View the article online for updates and enhancements.

You may also like

- Optimizing the classification ability of CNN for SAR fully polarized radar data based on DCGAN
  ZeCong Bu, Yue Zhang and AnMai Zheng

- Feasibility of CycleGAN enhanced low dose CBCT imaging for prostate radiotherapy dose calculation
  Y Chan, M Li, K Parodi et al.

- Conditional GAN Empowered Channel Estimation and Pilot Design for Reconfigurable Intelligent Surfaces assisted Symbiotic Radio System
  Yuan Zhou, Kaitian Cao, Lianning Cai et al.

This content was downloaded from IP address 122.163.64.247 on 10/07/2023 at 06:13

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
4 January 2023

REVISED
16 March 2023

ACCEPTED FOR PUBLICATION
21 March 2023

PUBLISHED
14 April 2023

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

ICEGAN: inverse covariance estimating generative adversarial network

Insoo Kim[1], **Minhyeok Lee**[2,*] and **Junhee Seok**[1,*]

[1] School of Electrical Engineering, Korea University, Seoul, Republic of Korea
[2] School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Republic of Korea
[*] Authors to whom any correspondence should be addressed.

E-mail: mlee@cau.ac.kr and jseok14@korea.ac.kr

Keywords: GAN, network estimation, CycleGAN, Monte-Carlo

Supplementary material for this article is available online

Abstract

Owing to the recent explosive expansion of deep learning, several challenging problems in a variety of fields have been handled by deep learning, yet deep learning methods have been limited in their application to the network estimation problem. While network estimation has a possibility to be a useful method in various domains, deep learning-based network estimation has a limitation in that the number of variables must be fixed and the estimation cannot be performed by convolutional layers. In this study, we propose a Generative Adversarial Network (GAN) based method, called Inverse Covariance Estimating GAN (ICEGAN), which can alleviate these limitations. In ICEGAN, the concepts in Cycle-Consistent Adversarial Networks are modified for the problem and employed to adopt gene expression data. Additionally, the Monte Carlo approach is used to address the fixed size in the network estimation process. Thus, sub-networks are sampled from the entire network and estimated by ICEGAN; then, the Monte Carlo approach reconstructs the entire network with the estimations. In the simulation study, ICEGAN demonstrated superior performances compared to conventional models and the ordinary GAN model in estimating networks. Specifically, ICEGAN outperformed an ordinary GAN by 85.9% on average when the models were evaluated using the area under curve. In addition, ICEGAN performed gene network estimation of breast cancer using a gene expression dataset. Consequently, ICEGAN demonstrated promising results, considering the deep learning-based network estimation and the proposed Monte Carlo approach for GAN models, both of which can be expanded to other domains.

1. Introduction

Sparse inverse covariance estimation is a task that estimates an adjacency matrix that describes the associations between variables from a covariance matrix. In the sparse inverse covariance matrix, the variables and associations between variables can be represented as nodes and edges in a sparse network, respectively. As a result, the sparse inverse covariance estimation is frequently considered synonymous with the sparse network estimation. These networks are applied in various disciplines and problems where the representation of interactions between variables is critical. For example, spares networks have been used for time-series networks containing economic factors [1] and various bioinformatic challenges, including time-to-event prediction and genome sequencing [2–5].

Numerous linear-based approaches have been investigated to estimate sparse networks [6, 7] and have incorporated penalties on weight parameters to account for the sparsity of networks. For example, the graphical lasso (GLASSO) approach, a lasso-based network estimation method that utilizes a covariance matrix, is the most often used linear-based network estimation method [6]. Additionally, the sparse partial correlation estimation (SPACE) approach and the PC algorithm are frequently used for the network estimation [8–10].

© 2023 The Author(s). Published by IOP Publishing Ltd

However, it is often challenging to apply such linear-based algorithms to a massive network with numerous variables, because the calculation time exponentially increases with the number of variables. Therefore, these approaches are rarely applicable to large databases, while the number of such databases has been expanding in recent years owing to the advancements in data technology [11]. For example, considering that human gene expression data often contain more than 20 000 genes [12, 13], these approaches are typically applied to only a few hundred genes due to the limitations [14].

This study proposes a novel solution to the problem using a deep-learning method called inverse covariance estimating generative adversarial network (ICEGAN). Numerous studies have sought to apply deep learning to various problems [15–18] because of recent advancements in deep learning models. Consequently, effective applications in image classification [19, 20], object detection [21, 22], a variety of natural language processing challenges, including language translation [23, 24] and text generation [25, 26], and financial portfolio management [27] have been developed.

The generative adversarial network (GAN) is a deep learning technique that has been extensively investigated in recent years [28]. Although GAN has been recently proposed in comparison to other deep learning models, it has already been used for a range of tasks, including image generation [29, 30], speech generation [31], and image translation [32, 33]. Additionally, by utilizing the adversarial learning process inherent in GAN, cycle-consistent GAN (CycleGAN) [32] effectively handles the problem of style transfer, which involves transferring an item to another style while retaining the primary properties of an object. CycleGAN can be used to create an image filter that converts a photograph into a piece of artwork [32].

The proposed model, ICEGAN, is inspired by the concept that the sparse network estimation problem can be viewed as a style transfer problem in which the covariance and the adjacency matrices represent two distinct styles of the same item, i.e. latent associations between variables. As a result, by employing covariance matrices as inputs, a CycleGAN model can address the problem of sparse network estimation. However, the standard CycleGAN model has a restriction in that both training and test data must be the same size. For instance, if a CycleGAN model is trained with 100 variables, which corresponds to a covariance matrix of size 100 × 100, the model can only estimate networks with 100 nodes. Consequently, each model for each number of variables must be constructed using the standard CycleGAN.

This problem is handled in ICEGAN using a Monte Carlo approach with a fixed-size GAN model to accommodate networks of specific sizes [34]. We trained a simple GAN model to convert a covariance matrix to an adjacency matrix. Then, a random subset of the network's variables is chosen, and the trained model estimates the edges between the subset of variables. This procedure was performed several times with the random variable; after that, the estimations for each edge were averaged.

In this study, we propose two novel concepts: (1) ICEGAN, which introduces style transfer algorithm in network estimation problem; (2) implementing Monte Carlo simulation methods in deep learning architecture. Here, ICEGAN and the Monte Carlo approach for the network estimation are examined using simulation and gene expression data. Latent networks and relationships between variables are hypothesized in the simulation study, and simulation data are then constructed according to the network. The ICEGAN model was assessed to determine whether it is capable of estimating the network using simulation data. Additionally, the model was tested against actual breast cancer gene expression data, which suggests an intriguing possibility.

2. Background

A GAN is a subtype of a deep learning model in which two adversarial neural networks are combined. During the training process, the minimax game is played between a generator and a discriminator. The objective of the generator is to produce realistic synthetic samples that closely resemble the input distribution from the known distribution. By contrast, the objective of the discriminator is to distinguish between fake and true samples. The target function below implements this minimax game between two adversarial agents:

$$Q = \min_{G}\max_{D}\sum\left(\log D(x) + \log\left(1 - D(G(z))\right)\right),$$

where D and G indicate the discriminator and generator, respectively, and z is the input for the generator, which is generally a Gaussian noise.

However, training a GAN using the conventional loss function is unstable and frequently fails to achieve Nash equilibrium, which is the global optimum of the generator and discriminator. Numerous techniques have been developed to address this shortcoming of typical GANs, including Wasserstein GANs [35] and spectral normalization (SN) [36].

In addition, when the GAN model is used to solve the style transfer problem, training distributions by the generator are substituted by the input style, and the generator is trained to convert the input style to the output style. As the baseline structure for the proposed model, we used a primitive version of the CycleGAN method suitable for style transfer using paired training data [33]. The objective function below implements the CycleGAN minimax game for the generator and discriminator:

$$Q^* = \min_G \max_D Q(G,D) + \lambda \mathbb{E}_{x,z}[\|x - G(z)\|_1],$$

where x denotes a sample in the target style, and z is a sample in the input style instead of Gaussian noise. This objective function attempts to minimize both the L1-loss of the generator and to the conventional target of the GAN.

3. Material and methods

3.1. Inverse covariance estimation

The inverse covariance estimation of the proposed model can be regarded as a style transfer problem in CycleGAN. A network can be represented by an adjacency matrix containing values of 0 and 1. In an adjacency matrix, a value of 1 between a row and column indicates a connection between the variables, thereby representing networks with the matrix. Generally, such an adjacency matrix can be estimated from a covariance matrix. Therefore, covariance matrices and adjacency matrices are considered as the two styles in a style transfer problem, which can be handled by CycleGAN.

Despite the similarity between the inverse covariance estimation and the style transfer problem with images, it is not suitable to handle the inverse covariance estimation with CycleGAN. The main difference between these problems is the neighborhood correlation in the sample matrix. While an image guarantees associations between neighboring pixels (i.e. neighbor elements in a matrix), a covariance matrix and adjacency matrix cannot define the neighborhood, and the order of the variable in those matrices is simply random or the order of observation. Thus, whereas convolutional neural networks are extensively used to address image applications [19, 20], the generator and discriminator in the proposed model use fully connected layers, which is in stark contrast to existing style transfer GAN models.

Generally, the proposed ICEGAN follows the concept of CycleGAN training. Specifically, the inverse covariance estimation by ICEGAN is performed using two independent generators and one discriminator. One generator converts the covariance matrix to the corresponding adjacency matrix, and the other generator converts the adjacency matrix to the covariance matrix. These steps are similar to the process of the CycleGAN. The discriminator was trained to classify pairs of fake and real samples. A pair of an estimated adjacency matrix and a real covariance matrix and a pair of a real adjacency matrix and an estimated covariance matrix are two types of pairs with fake samples. Using these pairs, the discriminator is trained to classify the generated samples as fake, and the generators aim to be determined as real. Pairs of estimated covariance matrices and estimated adjacency matrices are not used for training since the generators cannot be trained with such a pair.

Additionally, a cycle-consistency loss was employed in ICEGAN. In the cycle-consistency loss, the estimation was performed twice with the two generators in ICEGAN. This indicates that the covariance matrix passes sequentially through the two generators. Then, the estimated covariance matrix with these processes should have the same values as the original covariance matrix. Cycle-consistency loss utilizes such a concept and aims to minimize the differences between these two matrices. In addition, the cycle-consistency between an adjacency matrix and its estimation can be calculated using the same process. These cycle-consistency losses were used to train the two generators in ICEGAN.

Furthermore, a feature match loss, which corresponds to the concept of pixel loss in the Pix2Pix model, which is a primitive model of the CycleGAN, was used for the training of the generators. The feature match loss is the L1 loss between the estimated adjacency matrices and the real adjacency matrices. Although the generator can be trained solely with the feature match loss, applying the regulation in the connection of the network demonstrated superior results in inverse covariance estimation [6]. While the feature match loss generally forces the estimated adjacency matrices to be the same as real adjacency matrices, the errors indicated by the noises in the observations and covariance matrices are handled through the adversarial and cycle-consistency loss as the Pix2Pix model has applied adversarial GAN loss to sharpen the transferred image [33]. This process was also conducted using the estimated covariance matrices and real covariance matrices with the other generator.

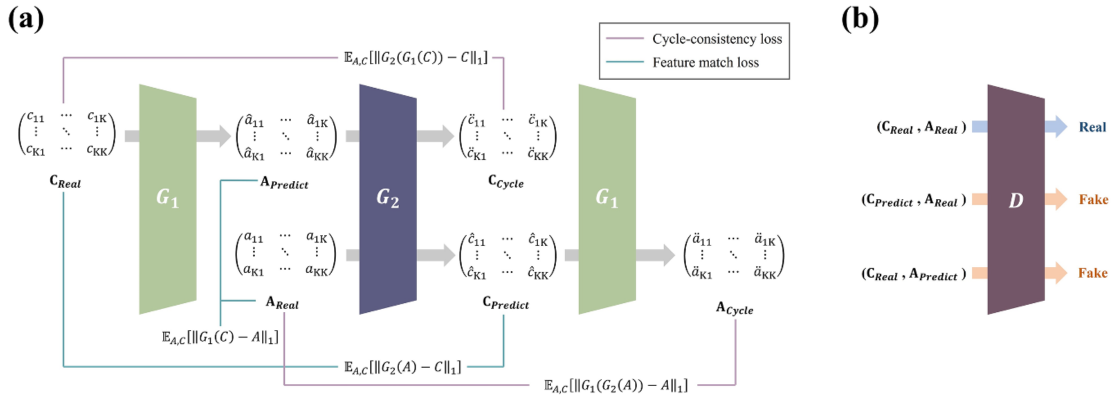
(a)
(b)
Cycle-consistency loss
Feature match loss
$\mathbb{E}_{A,C}[\|G_2(G_1(C)) - C\|_1]$
C_{Real}
G_1
$A_{Predict}$
G_2
C_{Cycle}
G_1
A_{Real}
$C_{Predict}$
A_{Cycle}
$\mathbb{E}_{A,C}[\|G_1(C) - A\|_1]$
$\mathbb{E}_{A,C}[\|G_2(A) - C\|_1]$
$\mathbb{E}_{A,C}[\|G_1(G_2(A)) - A\|_1]$
(C_{Real}, A_{Real})
Real
$(C_{Predict}, A_{Real})$
D
Fake
$(C_{Real}, A_{Predict})$
Fake

Figure 1. The schematic of training process of (**a**) the generators and (**b**) the discriminator of the proposed model.

The detailed loss functions are shown in the equations below:

$$L_{G1} = -\mathbb{E}_{A,C}[D(G_1(C), C)] + \lambda_1 \mathbb{E}_{A,C}[\|G_1(C) - A_1\|] + \lambda_2 \mathbb{E}_{A,C}[\|G_2(G_1(C)) - C_1\|],$$

$$L_{G2} = -\mathbb{E}_{A,C}[D(A, G_2(A))] + \lambda_1 \mathbb{E}_{A,C}[\|G_2(A) - C_1\|] + \lambda_2 \mathbb{E}_{A,C}[\|G_1(G_2(A)) - A_1\|],$$

$$L_G = L_{G1} + L_{G2},$$

$$L_D = 2\mathbb{E}_{A,C}[RELU(1 - D(A,C))] + \mathbb{E}_{A,C}[RELU(1 + D(G_1(C), C))] + \mathbb{E}_{A,C}[RELU(1 + D(A, G_2(A)))],$$

where L_G and L_D represent the generator loss function and discriminator loss function, respectively, λ_1 and λ_2 are hyperparameters, A denotes an adjacency matrix, C is a covariance matrix, G_1 is the generator estimating an adjacency matrix from a covariance matrix, G_2 is the generator estimating a covariance matrix from an adjacency matrix, and D indicates the discriminator.

In this study, the values of λ_1 and λ_2 were set to 10 and 1, respectively. To stabilize the training of the generator and discriminator, the Wasserstein loss, hinge loss, and SN are employed [35–37], which are state-of-the-art methods for the GAN training stabilization. An overall schematic of the training process of the proposed ICEGAN is shown in figure 1.

The neural network architecture of the ICEGAN is elaborated in this section. The generator comprised fully connected layers with 100 input and output dimensions and three hidden layers with 1024 nodes each. The Rectifier Linear Unit (ReLU) function was used as an activation function for the hidden layers, and the tanh was used as an activation function for the output layer. In addition, the discriminator was composed of fully connected layers with 200 input dimensions, one output dimension, and two hidden layers with 2048 nodes each. The leaky ReLU function was used as an activation function for all the layers in the discriminator. The input and output matrices were flattened during training because there was no association between the neighboring variables in the matrices. The detailed description of the generator and the discriminator of the ICEGAN is shown in figure 2.

The Adam optimizer [38] was used for training the generators and discriminator of the ICEGAN. The learning rates of the Adam optimizer were set at 8×10^{-4} for the discriminator and 5×10^{-4} for the two generators, respectively. The β_1 and β_2 values of the Adam optimizer were fixed of both optimizers at 0.9 and 0.999, respectively. Moreover, the models were trained with a batch size of 256 and for a total of 30 000 epochs.

After the estimations by ICEGAN, the Monte-Carlo approach was used to estimate a full network. The proposed ICEGAN structure possessed a limitation that a fixed network size is required. In this study, this issue is addressed using the Monte–Carlo approach with sub-networks sampled from an entire network. In particular, such sub-networks are handled by ICEGAN, and the entire network is then estimated using the Monte–Carlo approach. This process is elaborated upon in the following section.

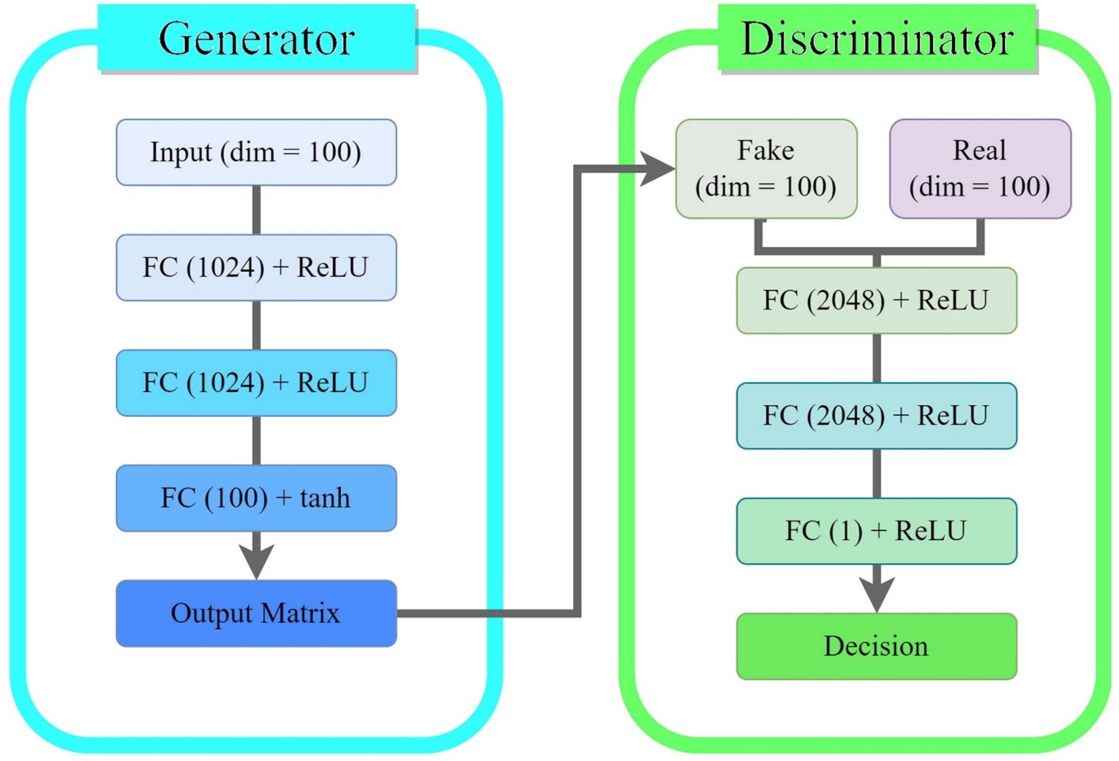

Figure 2. Detailed neural network architecture of the generator and discriminator of the ICEGAN.

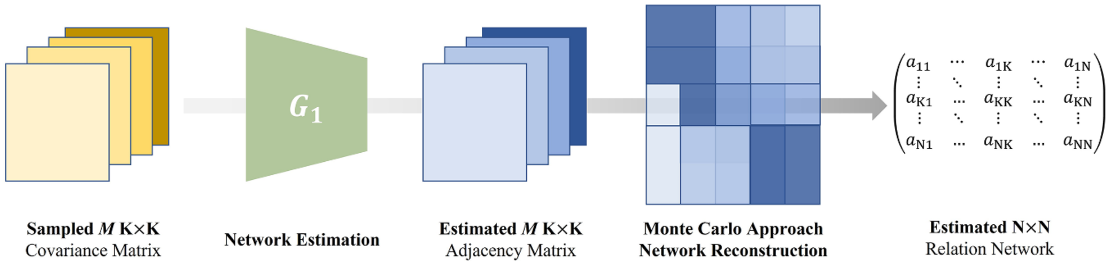
G_1
Sampled M K×K Covariance Matrix
Network Estimation
Estimated M K×K Adjacency Matrix
Monte Carlo Approach Network Reconstruction
a_{11} ⋯ a_{1K} ⋯ a_{1N}
a_{K1} ⋯ a_{KK} ⋯ a_{KN}
a_{N1} ⋯ a_{NK} ⋯ a_{NN}
Estimated N×N Relation Network

Figure 3. The schematic of an Monte–Carlo inference process of the ICEGAN.

3.2. Sampling and integration

The sampling and integrating part of the proposed method comprises a random covariance matrix sampling part and an estimated sampled adjacency matrix integration part. From the arbitrary size of the $N \times N$ covariance matrix, which is larger than the $K \times K$ matrix, K random nodes were selected to build the sampled covariance matrix, and the indexes of these nodes were memorized. For M iterations, $K \times K$ covariance matrix was sampled, and the $K \times K$ adjacency matrix was predicted from the covariance matrix for each iteration. The predicted edges were recorded in the index of the $N \times N$ adjacency matrix which will be the final prediction result of the proposed model. A schematic visualizing the actual inference step, which is the combined process of sampling and integration, is demonstrated in figure 3.

After M iterations, the probability of the edges was multiplied to the $N \times N$ adjacency matrix. For the sampled $K \times K$ matrix to be integrated into an arbitrary $N \times N$ adjacency matrix, at least N^2/K^2 iterations are required to calculate the original matrix without holes. Consequently, after M iterations, the average absolute value of the non-diagonal edges in the $N \times N$ adjacency matrix is MK^2/N^2 and the average absolute value of the diagonal edges is MK/N if the proposed inverse covariance estimator is optimal. For the sampled $K \times K$ matrix where $K = 10$ in the experiment, the probability of the diagonal edges was calculated as N/MK and the probability of the non-diagonal edges was calculated as N^2/MK^2 to normalize the absolute values of the edges to 1. The detailed process is analyzed in algorithm 1.

Algorithm 1. Monte Carlo simulation process.

```
Input: $N \times N$ covariance matrix C, Monte Carlo iteration number $M$, $K \times K$ adjacency matrix estimator G
Output: $N \times N$ adjacency matrix A
For $M$ iterations do
    Randomly sample $K$ indexes $a_1,\ a_2, \ldots a_k$ ($a_n < \mathrm{N}$)
    Build $K \times K$ covariance matrix I where $\mathrm{I}_{ij} = \mathrm{C}_{a_i a_j}$
    Estimate $K \times K$ adjacency matrix $\mathrm{A}'$, $\mathrm{G(I)} = \mathrm{A}'$
    Fill in the $N \times N$ adjacency matrix $\mathrm{A}_{a_i a_j} = \mathrm{A}'_{ij}$
Calculate the probabilities of edges $P_{ij}$
Multiply the probability to A, $\mathrm{A}_{ij} = P_{ij}\mathrm{A}_{ij}$
```

Through the process analyzed in algorithm 1, the final adjacency matrix was predicted from the input covariance matrix.

4. Results and discussion

4.1. Fixed size toy network inverse covariance estimation

To evaluate ICEGAN, we employed toy network examples. We randomly generated the sets of undirected networks of various sizes. A positive correlation between specific nodes is represented by 1 in the adjacency matrix, and a negative correlation is represented by -1. With this adjacency matrix representing a network, observations considering the linear correlation between nodes were generated with noise. The correlation matrix was calculated from the generated observations. Finally, pairs of adjacency and correlation matrices were used for ICEGAN. The algorithm and code used to generate the samples are presented in the supplementary.

The inverse covariance estimation part of the proposed framework is composed of three distinct adversarial models: two generators and one discriminator. The generators and discriminators of the proposed framework were trained using the two generated network sets. For each set, 1000 pairs of adjacency and covariance matrices were generated during the experiment. Out of the 1000 generated pairs, 950 pairs were used to train the proposed method and 50 pairs were used to verify the method. Typically, 950 samples were insufficient to train the model without accounting for overfitting. However, as noted in the Method section, the network is not related to the order of the nodes. Consequently, by shuffling the orders of the nodes, the training data can be augmented tremendously without changing the actual data. The detailed augmentation process is analyzed in algorithm 2.

Algorithm 2. Network data augmentation through the order shuffle.

```
Input: $K \times K$ adjacency matrix A, $K \times K$ covariance matrix C
Output: $K \times K$ adjacency matrix $\mathrm{A}'$, $K \times K$ covariance matrix $\mathrm{C}'$
For batch size B do
    Randomly shuffle $K$ indexes, $a_1,\ a_2, \ldots a_k$ $(1 \leqslant a_n \leqslant \mathrm{K})$
    Augment the $\mathrm{C}'$ from C, $\mathrm{C}'_{ij} = \mathrm{C}_{a_i a_j}$
    Augment the $\mathrm{A}'$ from A, $\mathrm{A}'_{ij} = \mathrm{A}_{a_i a_j}$
```

The proposed augmentation technique could theoretically generate K! different samples from the input matrices which are structurally the same isomorphic graphs. In the experiment, K was set as ten and 950 samples were augmented to approximately 3.4 billion samples.

Fifty pairs of matrices were tested to verify the generator that estimated the adjacency matrix from the covariance matrix, which was trained with the augmented training samples. The area under curve (AUC) calculated from the receiver operating characteristic (ROC) curve drawn from the true positive rate and false positive rate were implemented to test the performance of the proposed model. A comparison results of the proposed ICEGAN and linear-based methods is analyzed in figure 4. The AUC of the predicted network estimated from the proposed ICEGAN was 0.938 while the AUCs of GLASSO, SPACE, and PC algorithm were 0.869, 0.910, and 0.788 respectively, for the 10 × 10 networks with an average of two degrees. The AUC of the predicted network estimated from the ICEGAN was 0.888 while the AUCs of GLASSO, SPACE, and PC algorithm were 0.816 and 0.821, respectively, for the 10 × 10 networks with an average of three degrees. The AUC of the predicted network estimated from the ICEGAN was 0.985 while the AUCs of GLASSO and

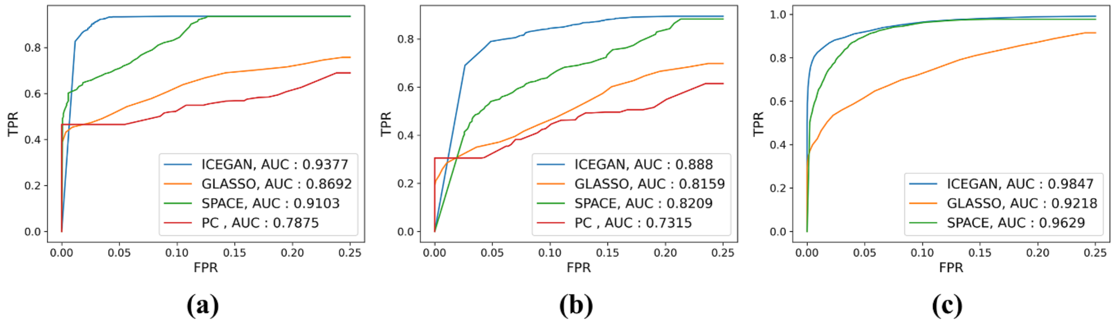

Figure 4. The ROC curves of the ICEGAN and linear based network estimation method in network with 10 variables and average degree 2 (a) and 3 (b) and with 20 variable of average degree 2 (c).

Table 1. The performance comparison results between the proposed method and other base line methods.

Method (AUC) Condition	ICEGAN	Vanilla GAN	Glasso	Space
Variable = 100, Edge = 1	0.9990	0.5998	0.9987	**0.9994**
Variable = 200, Edge = 1	**0.9994**	0.5990	0.9991	0.9993
Variable = 500, Edge = 1	**0.9997**	0.5957	0.9996	0.9984
Variable = 100, Edge = 2	**0.9875**	0.5175	0.9699	0.9870
Variable = 200, Edge = 2	**0.9918**	0.5175	0.9823	0.9837
Variable = 500, Edge = 2	**0.9956**	0.5138	0.9905	0.9855
Variable = 100, Edge = 4	**0.9164**	0.4655	0.8783	0.9015
Variable = 200, Edge = 4	**0.9395**	0.4597	0.9112	0.9149
Variable = 500, Edge = 4	**0.9595**	0.4601	0.9382	0.9217

The methods that showed the best performance were highlighted as bold.

SPACE were 0.922 and 0.963, respectively, for the 20 × 20 networks with an average of two degrees. The PC algorithm was computationally infeasible for networks with larger size than 10 × 10. The result implies that the proposed graph estimation method shows superior results compared to other linear-based methods in fixed-size graph estimation.

4.2. Arbitrary size toy network inverse covariance estimation

The results shown above verify the effectiveness of implementing ICEGAN in a fixed size network inverse covariance estimation. However, the trained generator of the proposed ICEGAN is only capable of estimating a fixed size network. As noted in the Method section, random sampling and integration of the input covariance matrix with the Monte Carlo simulation process are implemented in the proposed method to handle such limitations. Several networks with different sizes and degrees were estimated with the proposed model to show the effectiveness of the proposed network estimation model in an arbitrary size network estimation. Total nine different types of toy networks with three different network sizes and three different degrees were tested with the proposed method to show the novelty of the proposed method. Undirected networks with 100, 200, and 500 nodes with average 1, 2, and 4 edges were generated. Total 900 generated networks, 100 networks each for the proposed conditions were tested to verify the model. The AUC comparison results of the proposed method and other baseline methods under several conditions were employed as the performance evaluation metrics.

Similar to the fixed size network estimation, the proposed method is compared with the GLASSO and SPACE since the PC algorithm is computationally infeasible for the large networks. The performance of linear based methods decreases as the size of the network and the average degree of the network increase. The proposed ICEGAN showed at least comparable or improved performance compared to the linear based method in all 9 conditions. The detailed comparison results are presented in table 1. In particular, as shown in figure 5, when the network size was 100 and the number of average edges was 4, where the sparsity of the network was the lowest, the AUC value of the proposed ICEGAN was 0.9164, which is a superior result compared to linear based network methods. Since the performance of the linear based methods severely deprecates as the sparsity of the network decreases, the result implies that the proposed ICEGAN can be a more general solution for the network estimation problem.

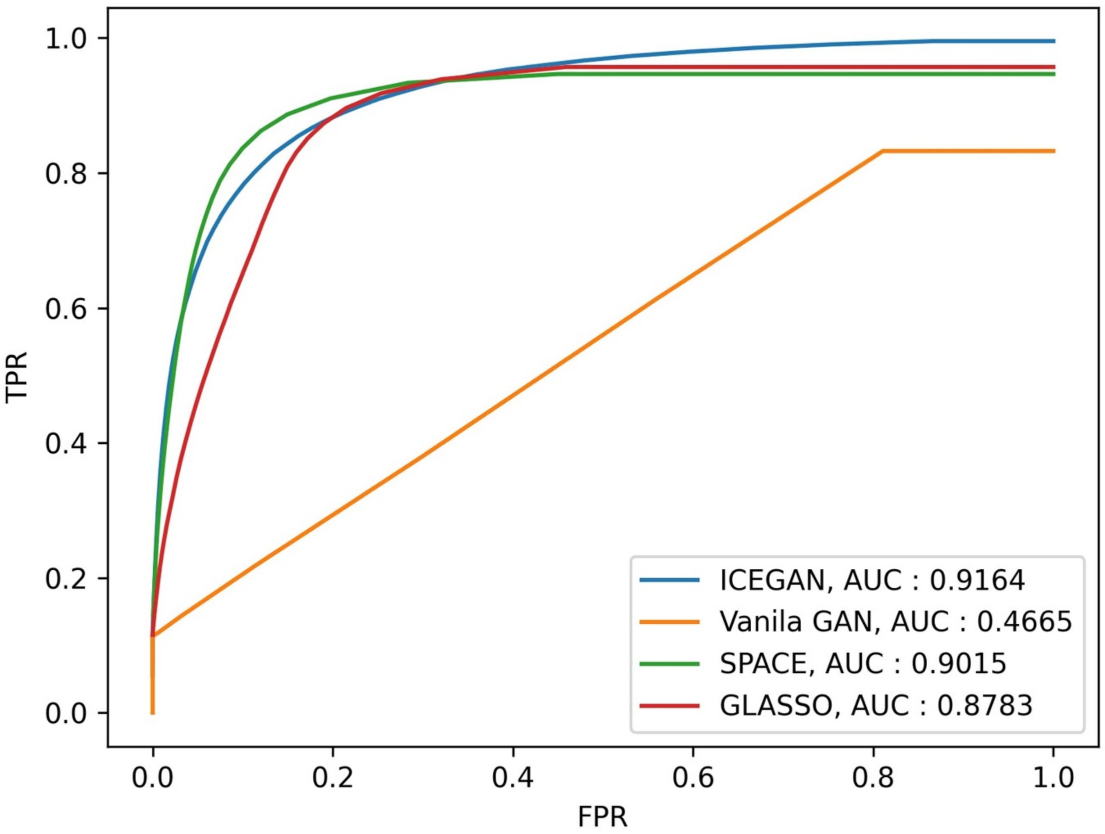

Figure 5. The ROC curve of the proposed method and other baseline methods with 100 variables and average 4 edges per variable.

4.3. Gene network estimation with ICEGAN

Although the proposed ICEGAN demonstrated novel results in network estimation problems, further verification of the method using a real-world dataset is indispensable. Gene network reconstruction from gene expression analysis is one of the real-world network estimation problem suitable for testing the proposed method [39]. Among several gene expression data, the gene expression database of the breast invasive carcinoma data (TCGA-BRCA) collected through The Cancer Genome Atlas projects [40] was used to verify the efficiency of ICEGAN. We implemented ICEGAN in TCGA-BRCA gene expression data to estimate the gene network from gene expression data. Among the 20 000 genes provided in TCGA data, we selected 55 genes found through numerous studies to be significantly associated with the breast cancer and used for network estimation in previous studies [35].

The gene relation network acquired with the proposed ICEGAN was investigated through the molecular signatures database (MSigDB) to compare the estimated gene network and annotated gene sets [41, 42]. The verification of the estimated gene network was held in a way whether the estimated network contains the real gene sets in MSigDB. Among several gene sets related to breast carcinoma, the HP_BREAST_CARCINOMA gene set, which is directly related to breast carcinoma was selected for the investigation [43]. The 22 genes in the HP_BREAST_CARCINOMA gene set overlapped with genes selected to construct the gene network. The estimated gene network and the gene set are shown in figure 6. The nodes in figure 6 are illustrated in green if the genes are contained in the HP_BREAST_CARCINOMA gene set; otherwise, they are shown in orange. All the edges of the network are the estimated edges of the network acquired with the ICEGAN.

As a result, *CENPA* was discovered as a hub gene that regulates the other genes, where nine edges were found to be connected with *CENPA*. This finding accorded with several related studies which argued that *CENPA* is one of the prognostic genes of breast cancer [44]. Additionally, *ATM* and *RAD50* were found that they are candidates of regulatory genes among HP_BREAST_CARCINOMA gene set. Since *ATM* and *RAD50* are well known to be significant genes for the prognosis of breast cancer [45, 46], previous studies supported the network estimated by ICEGAN. These results imply that the proposed ICEGAN can be employed for real-world data sets and proposes several possible biomedical findings.

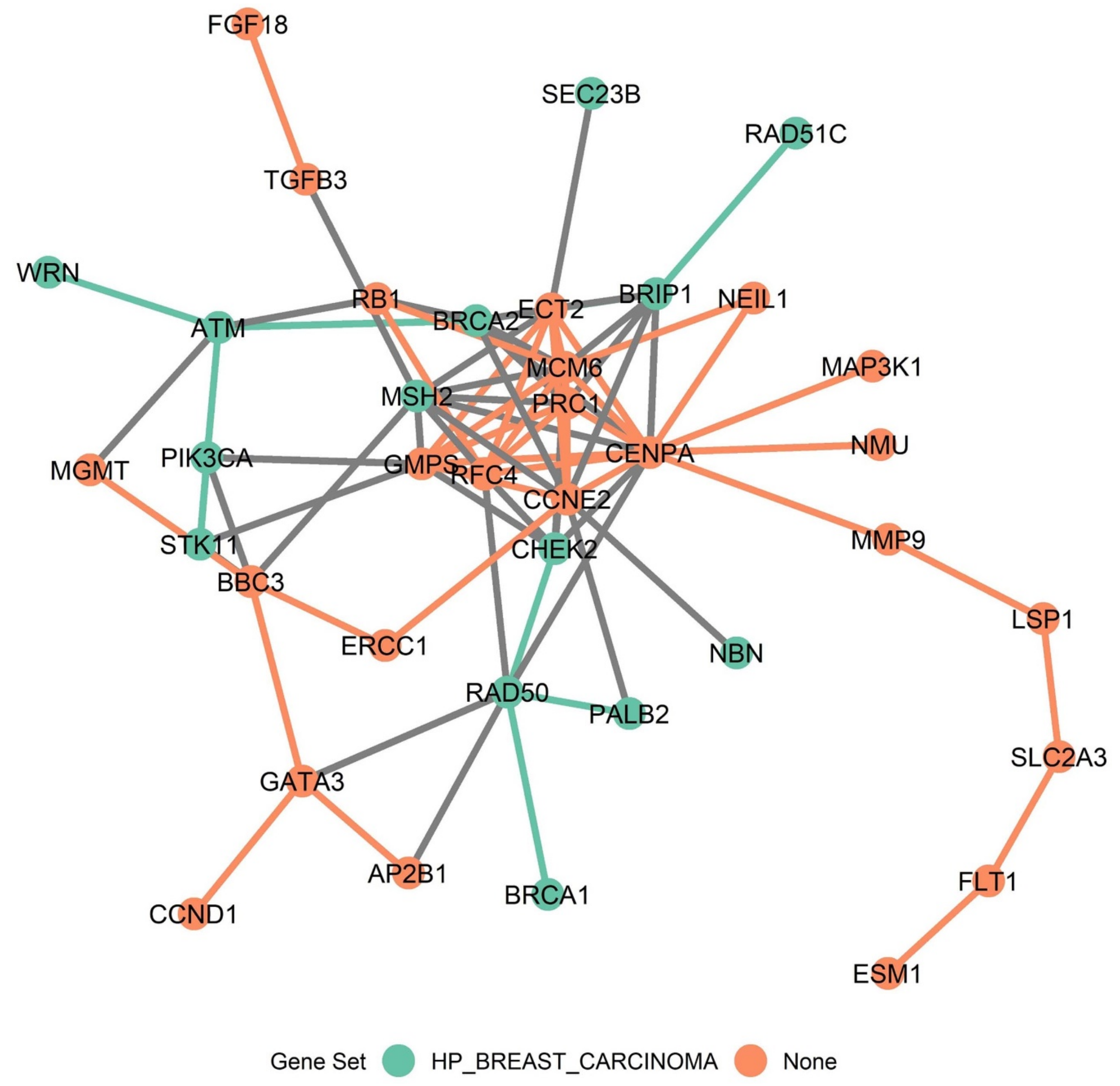

Figure 6. The gene network estimated from the breast cancer data through the ICEGAN. The green nodes and edges are the genes and relationships among the genes included in HP_BREAST_CARCINOMA gene set. The orange nodes and edges are the genes not included in the gene set.

5. Conclusion

In this study, we proposed the ICEGAN model which is an improved version of the GAN model that solves the network estimation problem as an extended version of an inverse covariance estimation problem. We implemented the Monte–Carlo approach in the GAN model to address the limitation of the fixed dimensionality of deep learning methods. By integrating the GAN model and the Monte-Carlo approach, the proposed ICEGAN successfully handled the arbitrary-size sparse network estimation problem in both simulation and real-world samples. The proposed ICEGAN demonstrated superior performance in estimating networks of the toy examples generated through simulations and constructing the gene network with the MSigDB gene set samples. Consequently, we hope this work to be the meaningful step toward the innovation in the field of deep learning.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work was supported by Samsung Electronics Co., Ltd (IO201214-08149-01) and a grant from the National Research Foundation of Korea (NRF-2022R1A2C2004003)

Conflict of interest

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID iDs

Insoo Kim https://orcid.org/0000-0001-6804-7129
Minhyeok Lee https://orcid.org/0000-0003-2562-172X
Junhee Seok https://orcid.org/0000-0002-6475-8457

References

[1] Barigozzi M and Brownlees C 2019 Nets: network estimation for time series *J. Appl. Econom.* **34** 347–64

[2] Finn R S, Qin S, Ikeda M, Galle P R, Ducreux M, Kim T-Y, Kudo M, Breder V, Merle P and Kaseb A O 2020 Atezolizumab plus bevacizumab in unresectable hepatocellular carcinoma *New Engl. J. Med.* **382** 1894–905

[3] Jones D T, Buchan D W, Cozzetto D and Pontil M 2012 PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments *Bioinformatics* **28** 184–90

[4] Xuan C, Wang Y, Zhang B, Wu H, Ding T and Gao J 2022 scBPGRN: integrating single-cell multi-omics data to construct gene regulatory networks based on BP neural network *Comput. Biol. Med.* **151** 106249

[5] Kim Y and Seok J 2020 Network estimation for censored time-to-event data for multiple events based on multivariate survival analysis *PLoS One* **15** e0239760

[6] Friedman J, Hastie T and Tibshirani R 2008 Sparse inverse covariance estimation with the graphical lasso *Biostatistics* **9** 432–41

[7] Danaher P, Wang P and Witten D M 2014 The joint graphical lasso for inverse covariance estimation across multiple classes *J. R. Stat. Soc.* B **76** 373

[8] Peng J, Wang P, Zhou N and Zhu J 2009 Partial correlation estimation by joint sparse regression models *J. Am. Stat. Assoc.* **104** 735–46

[9] Kalisch M and Bühlman P 2007 Estimating high-dimensional directed acyclic graphs with the PC-algorithm *J. Mach. Learn. Res.* **8** 613–36

[10] Harris N and Drton M 2013 PC algorithm for nonparanormal graphical models *J. Mach. Learn. Res.* **14** 3365–83

[11] Wang W, Zhang M, Chen G, Jagadish H, Ooi B C and Tan K-L 2016 Database meets deep learning: challenges and opportunities *ACM SIGMOD Rec.* **45** 17–22

[12] Edgar R, Domrachev M and Lash A E 2002 Gene expression omnibus: NCBI gene expression and hybridization array data repository *Nucleic Acids Res.* **30** 207–10

[13] Brazma A and Vilo J 2000 Gene expression data analysis *FEBS Lett.* **480** 17–24

[14] Emilsson V, Thorleifsson G, Zhang B, Leonardson A S, Zink F, Zhu J, Carlson S, Helgason A, Walters G B and Gunnarsdottir S 2008 Genetics of gene expression and its effect on disease *Nature* **452** 423–8

[15] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P and Garcia-Rodriguez J 2018 A survey on deep learning techniques for image and video semantic segmentation *Appl. Soft Comput.* **70** 41–65

[16] Sezer O B, Gudelek M U and Ozbayoglu A M 2020 Financial time series forecasting with deep learning: a systematic literature review: 2005–2019 *Appl. Soft Comput.* **90** 106181

[17] Seo J, Kim I and Seok J 2023 Grid-wise simulation acceleration of the electromagnetic fields of 2D optical devices using super-resolution *Sci. Rep.* **13** 435

[18] Kim I, Park S J, Jeong C, Shim M, Kim D S, Kim G-T and Seok J 2022 Simulator acceleration and inverse design of fin field-effect transistors using machine learning *Sci. Rep.* **12** 1140

[19] Tan M and Le Q 2019 Efficientnet: rethinking model scaling for convolutional neural networks *Int. Conf. on Machine Learning, PMLR* pp 6105–14

[20] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25 pp 1097–105

[21] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: unified, real-time object detection *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 779–88

[22] Zhao Z-Q, Zheng P, Xu S-T and Wu X 2019 l. systems, object detection with deep learning: a review *IEEE Trans. Neural Netw.* **30** 3212–32

[23] Devlin J, Chang M-W, Lee K and Toutanova K 2018 BERT: pre-training of deep bidirectional transformers for language understanding (arXiv:1810.04805v2)

[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* vol 30 pp 5998–6008

[25] Fedus W, Goodfellow I and Dai A M 2018 Maskgan: better text generation via filling in the_ (arXiv:1801.07736)

[26] Kim Y, Lee J H, Choi S, Lee J M, Kim J-H, Seok J and Joo H J 2020 Validation of deep learning natural language processing algorithm for keyword extraction from pathology reports in electronic health records *Sci. Rep.* **10** 1–9

[27] Yun H, Lee M, Kang Y S and Seok J 2020 Portfolio management via two-stage deep learning with a joint cost *Expert Syst. Appl.* **143** 113041

[28] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* vol 27 pp 2672–80

[29] Brock A, Donahue J and Simonyan K 2018 Large scale GAN training for high fidelity natural image synthesis (arXiv:1809.11096v2)

[30] Lee M and Seok J 2019 Controllable generative adversarial network *IEEE Access* **7** 28158–69

[31] Pascual S, Bonafonte A and Serra J 2017 SEGAN: speech enhancement generative adversarial network (arXiv:1703.09452)

[32] Zhu J-Y, Park T, Isola P and Efros A A 2017 Unpaired image-to-image translation using cycle-consistent adversarial networks *Proc. IEEE Int. Conf. on Computer Vision* pp 2223–32

[33] Isola P, Zhu J-Y, Zhou T and Efros A A 2017 Image-to-image translation with conditional adversarial networks *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 1125–34

[34] Fong S J, Li G, Dey N, Crespo R G and Herrera-Viedma E 2020 Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction *Appl. Soft Comput.* **93** 106282

[35] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks *Int. Conf. on Machine Learning, PMLR* pp 214–23

[36] Miyato T, Kataoka T, Koyama M and Yoshida Y 2018 Spectral normalization for generative adversarial networks (arXiv:1802.05957v1)

[37] Lim J H and Ye J C 2017 Geometric gan (arXiv:1705.02894v2)

[38] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980v9)

[39] Chakrapani H B, Chourasia S, Gupta S and Haldar R 2021 Effective utilisation of influence maximization technique for the identification of significant nodes in breast cancer gene networks *Comput. Biol. Med.* **133** 104378

[40] TCGA TCGA-BRCA TCGA research network (available at: www.cancer.gov/tcga)

[41] Subramanian A, Tamayo P, Mootha V K, Mukherjee S, Ebert B L, Gillette M A, Paulovich A, Pomeroy S L, Golub T R and Lander E S 2005 Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles *Proc. Natl Acad. Sci.* **102** 15545–50

[42] Liberzon A, Subramanian A, Pinchback R, Thorvaldsdóttir H, Tamayo P and Mesirov J P 2011 Molecular signatures database (MSigDB) 3.0 *Bioinformatics* **27** 1739–40

[43] Köhler S, Gargano M, Matentzoglu N, Carmody L C, Lewis-Smith D, Vasilevsky N A, Danis D, Balagura G, Baynam G and Brower A M 2021 The human phenotype ontology in 2021 *Nucleic Acids Res.* **49** D1207–7

[44] Zhang S, Xie Y, Tian T, Yang Q, Zhou Y, Qiu J, Xu L, Wen N, Lv Q and Du Z 2021 High expression levels of centromere protein A plus upregulation of the phosphatidylinositol 3–kinase/Akt/mammalian target of rapamycin signaling pathway affect chemotherapy response and prognosis in patients with breast cancer *Oncol. Lett.* **21** 1–12

[45] Stucci L S, Internò V, Tucci M, Perrone M, Mannavola F, Palmirotta R and Porta C 2021 The ATM gene in breast cancer: its relevance in clinical practice *Genes* **12** 727

[46] Heikkinen K, Rapakko K, Karppinen S-M, Erkko H, Knuutila S, Lundán T, Mannermaa A, Børresen-Dale A-L, Borg Å and Barkardottir R B 2006 RAD50 and NBS1 are breast cancer susceptibility genes associated with genomic instability *Carcinogenesis* **27** 1593–9