One-class SVM to identify candidates to reference genes based on the augment of RNA-seq data with Generative Adversarial Networks

Edwin J. Rueda[1][0000−0003−2818−1555], Rommel Ramos[1,2][0000−0002−8032−1474], Edian Franco[2][0000−0001−9715−9437], Orlando Belo[3][0000−0003−2157−8891], and Jefferson Morais[1][0000−0002−8566−3238]

[1] Department of Computer Science, Federal University of Para, Belem, Brazil
edwin.rojas@icen.ufpa.br, jmorais@ufpa.br
[2] Institute of Biological Sciences, Federal University of Para, Belem, Brazil
rommelthiago@gmail.com
[3] University of Minho, Braga, Portugal
obelo@di.uminho.pt

Abstract. Reference genes (RG) are constitutive genes required for the maintenance of basic cellular functions. Different high-throughput technologies are used to identify these types of genes, including RNA sequencing (RNA-seq), which allows measuring gene expression levels in a specific tissue or an isolated cell. In this paper, we present a new approach based on Generative Adversarial Network (GAN) and Support Vector Machine (SVM) to identify *in-silico* candidates for reference genes. The proposed method is divided into two main steps. First, the GAN is used to increase a small number of reference genes found in the public RNA-seq dataset of *Escherichia coli*. Second, a one-class SVM based on novelty detection is evaluated using some real reference genes and synthetic ones generated by the GAN architecture in the first step. The results show that increasing the dataset using the proposed GAN architecture improves the classifier score by 19%, making the proposed method have a *recall* score of 85% on the test data. The main contribution of the proposed methodology was to reduce the amount of candidate reference genes to be tested in the laboratory by up to 80%.

Keywords: Generative adversarial networks · novelty detection · one-class svm · RNA-seq · reference genes.

1 Introduction

Reference genes (RG) are constitutive genes required for the maintenance of basic cellular function. These genes are expressed in all cells of an organism under normal and abnormal conditions and are used in internal controls in investigations of gene expression analysis because their level of expression remains relatively constant in different cells, tissues, and stress levels [6] [7] [14]. For instance, in the real-time reverse transcription-polymerase chain reaction (RT-qPCR) which provides accurate and reproducible quantification of genetic copies, the selection of RG is essential for accurate normalization of gene expression data obtained [10] [12].

The RG identification is not a trivial task because in some tissues or cells, the number of validated RG in the literature is small, and the level of expression of some of these genes varies depending on tissue, cell, or level of stress. Several methods to identify RG are available in the literature, mainly using optimization algorithms and unsupervised machine learning (ML) approach [1] [3] [11] [13].

In [3] is proposed a methodology based on clustering algorithms using Euclidean distance as a similarity metric. In this methodology, the number of clusters is chosen based on the SD validity index and the Dunn index [8]. Then, based on some reference genes, are selected the clusters in which at least one reference gene is present, reducing the number of candidate genes to be considered. Finally, the candidate genes are selected based on the coefficient of variation and the standard deviation of the expression level of each gene.

In [1] is proposed a methodology based on optimization algorithms, more specifically a particular type of linear programming called minimum cost network flow problem. The idea is to find the cheapest path in a directed, acyclic n_s-dimensional graph, where n_s denotes the total number of samples, from the lowliest, non-zero expressed gene (source) to the most highly expressed gene (sink), given several constraints. The genes with identically zero expression are omitted from the analysis and the distance from one gene to another is calculated using the Euclidean distance. Finally, the genes belonging to the cheapest path are considered candidate genes.

The main disadvantage of current methods is based on the small set of reference genes available in the literature for some tissues or cells. Therefore, methods that use the Euclidean distance between unclassified genes and reference genes as a similarity metric to detect potential candidates are not giving importance to unclassified genes that are distant from the reference genes, but that could be candidates due to stability in their gene expression.

This paper proposes a novel approach based on Generative Adversarial Networks (GAN) and a one-class classifier based on novelty detection to the problem of reference genes identification. The proposal is divided into two main steps. First, the GAN is used to generate enough synthetic reference genes from the actual reference genes available in the literature for the Escherichia coli MG1655 dataset [10] [1]. Second, a one-class classifier is trained using some real reference genes and synthetic ones generated by the GAN in the first step to detect new candidate genes. In this work, we adopted the support vector machine as a one-class classifier, since this classifier has a good performance in the literature [9] [15].

The remaining sections of this paper are organized as follows. Section II shows the main related works and the contributions of this work; Section III presents the proposed method for candidate reference genes identification, and the results are discussed in Section IV. Finally, Section V shows the final remarks on this research.

2 Related Work

Different papers in the literature propose the *in-silico* identification of candidates for reference genes. These approaches are based on datasets that contain a minority of genes labeled as reference genes. The objective is to classify the remaining genes as candidates or not to reference genes.

Currently, emerging different methods that use optimization algorithms and unsupervised machine learning algorithms to tackle the problem. In [1] was proposed a novel method for the identification of reference genes called $moose^2$. This method uses the minimum cost network flow problem to find candidates for RG. The idea is to find the cheapest path in a directed, acyclic n_s-dimensional graph, where n_s denotes the total number of samples, from the lowliest, non-zero expressed gene (source) to the most highly expressed gene (sink), taking into account that for any edge (i, j) that connects two nodes (genes) in the graph, this edge is in the direction of the lower to the higher ranked genes, where the ranking is determined by sorting the values of the gene expression averaged over all samples. Note that in the graph, each gene is connected to every other gene. The data set used is *Escherichia coli* obtained from CGSC (Coli Genetic Stock Center) at Yale University (http://cgsc.biology.yale.edu), where from 6 RG, $moose^2$ identifies 27 possible candidates for RG.

In [3] is proposed a methodology based on clustering algorithms using Euclidean distance as a similarity metric. In this methodology, the number of clusters is chosen based on the SD validity index and the Dunn index [8]. Then, based on some reference genes, are selected the clusters in which at least one reference gene is present, reducing the number of candidate genes to be considered. Finally, the candidate genes are selected based on the coefficient of variation and the standard deviation of the expression level of each gene. This method was evaluated in the *Corynebacterium pseudotuberculosis* strains CP1002 and CP25 database [16], obtaining 134 candidate genes in total.

Vandesompele et al. [13] presents the gNorm algorithm, which uses a measure of gene-stability to determine the stability of gene expression. For each control gene, they determine the variation in pairs with all other control genes as the standard deviation of the logarithmically transformed expression ratios and define the stability measure of the internal control gene M as the average variation in pairs of a particular gene with all other control genes. Genes with the lowest M values have the most stable expression. Therefore, they can be considered candidates for RG.

These current methods manage to identify candidate genes based on a number of initial RG, which is sometimes inefficient due to the small number of RG identified in certain cells or tissues. Therefore, in this paper, we propose to increase the number of reference genes synthetically through a GAN architecture.

Then, based on the augmented reference data, train a one-class SVM classifier to select from a dataset of unclassified genes, the possible candidates for RG.

3 Proposed Method

This section presents the dataset used to select the possible candidates for RG and the algorithms used in this approach. Fig. 1 summarizes the proposed method which consists of two main steps. First, the amount of reference genes available in the dataset is increased using a proposed GAN architecture. Second, a one-class classifier based on support vector machine (SVM) is trained to detect new candidates for reference genes using 70% of the real reference genes and synthetic genes generated by the GAN architecture in the first step. The performance of the proposed classifier is evaluated with 30% of the remaining reference genes available in the training set.

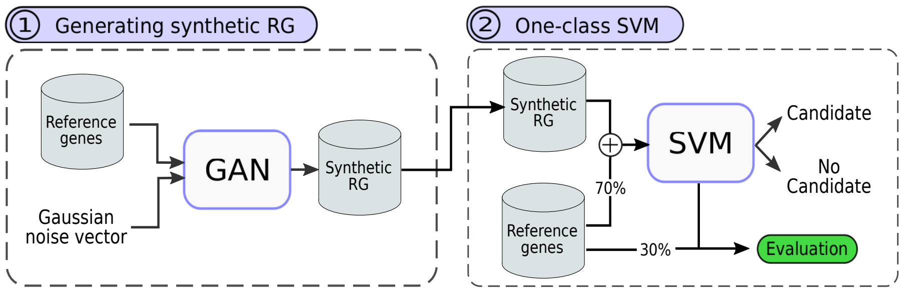


Fig. 1. The proposed methodology for the selection of candidate genes for reference genes.

The next subsection explains the dataset used, its initial processing for its subsequent increase through the proposed GAN architecture. Then, the one-class SVM algorithm used for the selection of candidates is explained.

3.1 Dataset

This work uses the RNA-seq database of *Esquerichia coli* MG1655 available at [1]. This dataset consists of 4293 genes that were subjected to stress conditions. Three different samples were removed at 0, 30, and 90 minutes, and immediately mixed with 0.25 vol of RNA stop solution (95 % ethanol, 5 % phenol) on ice. Note that the dataset has 9 features, 3 for each sampling time.

For the identification of the largest number of RG in this dataset, a review of the literature was necessary, which consisted of taking the genes that were at least tested in three different studies. Finding 21 reference genes in total, 15 in [10] and 6 in [1].

3.2 Data processing

The RNA-Seq dataset was normalized by reads per kilobase of exon model per million mapped reads (RPKM) and applying $(log_2 + 1)$ to mitigate the number of outliers and increase the proximity of the data features. To filter the dataset and based on the fact that the reference genes have a minimum level of variation [2], a stability test is performed based on the coefficient of variation (CV) given by Equation 1, where σ represents the standard deviation of the set of features of each gene, and μ represents its mean.

$$CV = \frac{\sigma}{\mu} \tag{1}$$

The stability test consists of selecting the genes for which their CV value is within an interquartile range given by $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$ with $k = 1.5$. Note that genes with a CV value outside this range are not considered. Finally, the data is scaled between $[-1, 1]$ as show in Equation 2, where X_t represents the transformed data, x_i the *i-th* gene and X_{max} and X_{min} represent the maximum and minimum values for each features of the dataset.

$$X_t = \frac{2(x_i - X_{min})}{X_{max} - X_{min}} - 1 \tag{2}$$

3.3 Generative Adversarial Nets

Generative adversarial networks were introduced by [5]. The concept was developed for the estimation of generative models based on adversarial processes. In these adversarial processes are trained two neural networks; the first, a generative network $G\left(z, \theta_g\right)$ which receives as input some noise variables from a, $p_z\left(z\right)$ distribution and aims to learn a p_g distribution on training data x, and a second discriminatory network $D\left(x, \theta_d\right)$, which has as output a single scalar which represents the probability that x comes from the distribution of the real data p_{data}, and not from p_g. In this minimax game, there is only one solution, which occurs when the G network recovers the distribution of the real data p_{data}, making the D network fail to distinguish if a data comes from the original distribution or the distribution generated by the G network, $D\left(x\right) = \frac{1}{2}$.

Fig. 2 illustrates the training process of the GAN architecture. This training is divided into two main steps. First, the discriminative model D is trained based on real and synthetic data and its weights are frozen (these weights are not updated in the second step). Second, the generative model G is trained based on an input noise vector to maximize the probability that the model D makes a mistake. After repeated steps one and two a defined amount of epochs, the model G may generate synthetic data similar to the original data.

The objective of this approach is to generate enough synthetic RG with a GAN architecture to be able to build a one-class model as general as possible. For this, Fig. 3 illustrates the proposed GAN architecture. Where the generator model (Fig. 3(a)) takes as input a noise vector based on a normal distribution

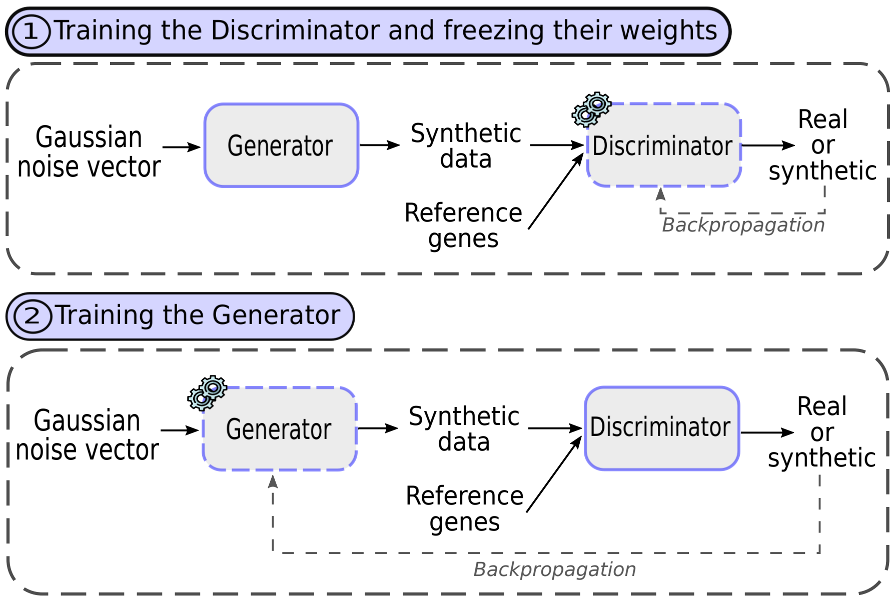
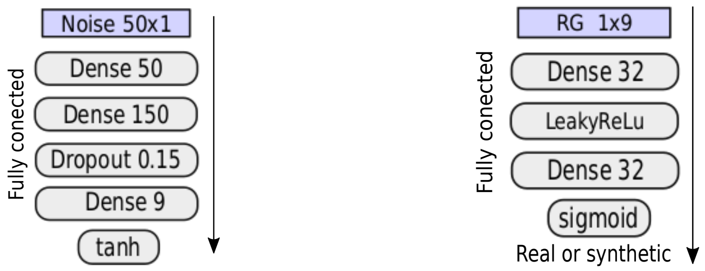


Fig. 2. The learning process of the Generative Adversarial Nets.

with $\mu = 0$ and $\sigma = 1$ and produces as output a synthetic reference gene. On the other hand, the discriminator model (Fig. 3(b)) was also built with fully connected dense layers. This model takes as input real reference genes (real distribution) and synthetic reference genes (generated by the G network) and its output represents the probability that a gene belongs to the real data distribution. The number of layers and neurons was taken based on the lowest cost value for the metrics proposed in Equations 9 and 10.



(a) Generator model. (b) Discriminator model.

Fig. 3. The GAN architecture.

The GAN architecture was trained with the *stochastic gradient descent* algorithm with a different learning rate l_r for the discriminator and the generator. These learning rates decreased at each epoch by a factor equal to $\frac{l_r}{epochs}$ to improve the convergence of the GAN. The initial weights for each layer were chosen based on *Xavier uniform initializer* [4], which initializes the weights based on a uniform distribution $U[-t,t]$, where t is equal to $\sqrt{6/\left(n_j+n_{j+1}\right)}$ being n_j the number of input neurons and n_{j+1} the number of output neurons in the j layer.

3.4 One-Class SVM

The one-class support vector machine induced by [17], is an algorithm that tries to estimate a function f, being $f \geq 0$ for positive examples and $f < 0$ for any other example. In this strategy, the data is mapped with a function Φ in a new characteristic space F and separated from the origin with a maximum margin.

Consider the training set $x_1, \ldots, x_\ell \in \mathcal{X}$, where $\ell \in \mathbb{N}$ is the number of observations, and $\mathcal{X}$ is some set $\in \mathbb{R}^N$. To separate the training set from the origin, the following quadratic problem is solved:

$$\min_{w\in F,\xi\in\mathbb{R}^\ell,\rho\in\mathbb{R}} \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho \tag{3}$$

$$s.t. f\left(x\right) = \left(w \cdot \Phi\left(x_i\right) - \rho\right) \geq -\xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, \ell \tag{4}$$

Deriving the dual problem, the solution can be shown to have an support vector expansion:

$$f\left(x\right) = sgn\left(\sum_i \alpha_i k\left(x_i, x\right) - \rho\right) \tag{5}$$

where the *sgn* function is equal to 1 for values greater than or equal to zero and -1 otherwise. Patterns x_i with nonzero α_i are called support vectors, where the coefficients are found by solving the dual problem:

$$\min_\alpha \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k\left(x_i, x_j\right)\ \ s.t\ 0 \leq \alpha_i \leq \frac{1}{\nu\ell},\ \sum_i \alpha_i = 1. \tag{6}$$

In this approach, the parameter ν is a trade-off between the normal and anomaly data in the dataset, which was set to 0.010452, and the mapping function Φ was represented by the Gaussian kernel given in Equation (7), where γ is equal to $1/n_{fe}$ and n_{fe} is the number of features of the training set.

$$k\left(x, y\right) = e^{-\gamma\|x-y^2\|} \tag{7}$$

If the proposed dual problem is solved, the decision function $f\left(x\right)$ will take positive values for the greatest number of samples x_i contained in the training set. To validate the performance of the proposed classifier and based on the premise that we only have a positive class (RG), we consider only the *recall* metric:

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

where TP (True Positives) represents the number of genes that the classifier correctly classified as RG, and FN (False Negatives) represents the amount of RG that the classifier classified as non-RG. This metric allows us to evaluate the ability of our classifier to recognize reference genes. Therefore, a value close to 1 indicates that the classifier has a good performance.

4 Experiments and Results

This section presents the experiments and results for the identification of candidates to RG based on the use of GAN to synthetically increase the training set, and based on this dataset, build a one-class SVM classifier. The focus of this study was based on the creation of the GAN architecture and the one-class SVM classifier for the identification of possible RG candidates in *Escherichia coli* MG1655 dataset obtained from [1].

For this study, the dataset was cleaned taking into account two important stages. In the first stage, we removed the genes for which their expression value was equal to zero. In the second stage, based on the CV (Equation 1), we eliminated all the genes for which the CV was not within the interquartile range (outliers). Table 1 shows the data processing to obtain the final dataset.

Table 1. Data processing stages

Number of initial genes	Number of genes expressed	Outliers based on the CV	Final Dataset
4293	4191	3	4188

Of the 21 RG identified in the literature for *Escherichia coli*, the *idnT* gene was not taken into account because it's CV was outside the interquartile range calculated for the reference genes. Note that were calculated two interquartile ranges, one based on the reference genes and the other based on the remaining genes. Table 2 shows the list of 20 RG used in this study.

Table 2. List of reference genes used in this study

Reference genes
cysG, hcaT, rrSA, ihfB, ssrA, gyrA, recA, rpoB, rpoA, gyrB, rho, ftsZ, secA, rpoC, gmk, adk, rpoD, dnaG, glnA, recF

4.1 Training the Generative Adversarial Network

For the training of the GAN architecture (Fig. 3), a parameter adjustment was made in the G and D networks to reach the global optimum ($D(x) = \frac{1}{2}$). For this reason, we opted to update the weights of both networks (θ_g and θ_d) based on all available reference genes ($batch\_size = 20$), because this allowed the network to converge faster. Note that the weights of network D are updated based on 20 reference genes and 20 synthetic genes generated by network G. Finally, 1700 epochs were used for the training of the proposed GAN architecture, considering that an epoch occurs when all the genes are iterated in the training set (one *batch_size*). The parameters for training the GAN architecture are presented in Table 3.

Table 3. Parameters used for training the GAN architecture

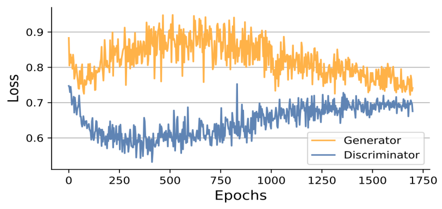
Parameter	Generator	Discriminator
Optimizer	SGD	SGD
Learning rate	0.00015	0.001
Decay rate	0.00015/1700	0.001/1700
Momentum	0.92	0.9
Epochs	1700	1700

The selection of the best initial weights taking the *Xavier uniform initializer* was based on a similarity metric $S(x, x')$ proposed in Equation (9), where x represents the training set given by a matrix $\in \mathbb{R}_{(20,9)}$, x' represents the set of synthetic data generated by the G network, and $\hat{y}$ represents the class predicted by network D for synthetic data, which takes values equal to 1 for real data and 0 for synthetic data. The parameters n_f, n_g, and m represent the number of characteristics of each gene, the number of synthetic genes, and the number of RG in the training set. Note that $x_i^{(k)}$ represents the feature k in the *i-th* gene.

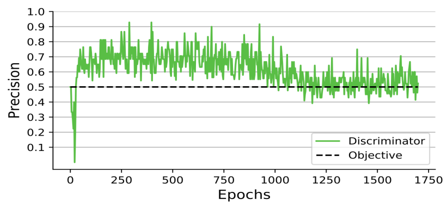
$$S\left(x, x'\right) = \sum_{i}^{m} \sum_{j}^{n_g} \sum_{k}^{n_f} \frac{|x_i^{(k)} - x'^{(k)}_j|}{n_f n_g m} + |0.5 - \frac{1}{n_g} \sum_{j}^{n_g} \hat{y}_j| \qquad (9)$$

The GAN network was trained on a hundred different occasions, for each of these occasions, the average of the similarity $S(x, x')$ was calculated based on 30 repetitions, wherein each repetition, were generated 300 synthetic genes. Thus, we chose the weights for which the GAN architecture presented the lowest similarity value.

Fig. 4 shows the convergence process in the training of the proposed GAN architecture, where Fig. 4(a) shows that the loss (based on binary cross_entropy) tends to 0.7 and Fig. 4(b) shows that the accuracy of the Discriminator tends to 0.5 (global optimum). These results indicate that network G is generating synthetic genes similar to the real ones, making network D unable to distinguish them.



(a) Loss in GAN architecture.



(b) Precision in GAN architecture.

Fig. 4. Convergence process in the training of the GAN architecture.

Principal component analysis PCA [18] is used for the graphical representation of the augmented dataset, with this representation, we can visually observe the similarity between the original and the synthetic data generated by the G network. For the choice of the best set of synthetic data, we generate 5000 different sets of 300 synthetic genes and based on the metric $E(x')$ proposed in Equation (10), where CV represents the coefficient of variation for the *i-th* synthetic gene, we chose the set of synthetic genes with the lowest value of this metric ($E = 0.11$). Figure 5 shows the 2-PCA representation of the RG (denoted by a green star) and the synthetic genes (denoted by a red circle), where we can see that the genes generated by the GAN architecture are not distant from the real RG.

$$E(x') = \frac{1}{n_g} \sum_{j}^{n_g} \left[CV(x'_j) + \frac{1 - D(x'_j)}{D(x'_j)} \right] \tag{10}$$

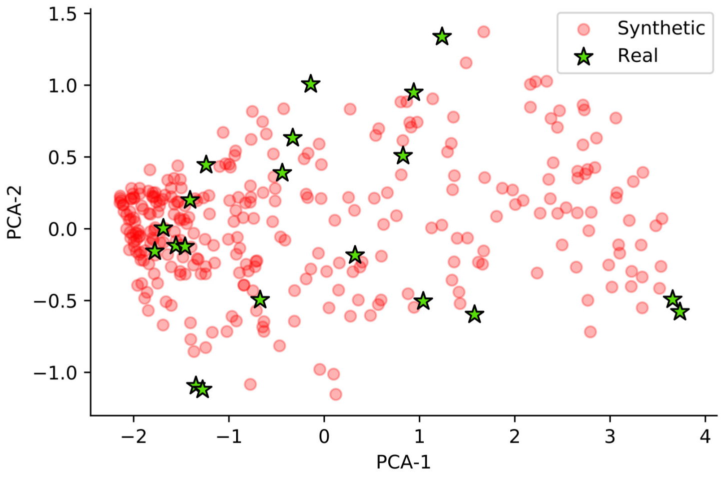


Fig. 5. Principal component analysis between real and synthetic data.

4.2 Selection of candidate genes

The selection of candidates to RG is based on the one-class SVM classifier. We train the proposed classifier based on the augmented training set that was built based on 70% of the RG and the 300 synthetic genes generated in the previous step.

The validation of this classifier (Equation (8)) was performed with four sets of data. Table 4 shows the results of the proposed classifier for each of these data sets. Where in the training set, we calculate two different *recalls*, the first for the augmented dataset, which consists of the augmented data used for classifier training, and the second based only in 70% of the RG chosen above. For the validation of the classifier in the test data, these were divided into two sets, the first set with the remaining 30% of the RG, and the second set based only on new synthetic RG generated by the proposed GAN architecture.

Table 4. Result of the recall metrics in the different datasets

Metrics	Training data		Test data	
	Augmented data	Reference genes (70%)	Reference genes (30%)	Synthetic data
Recall	98.40%	92.85%	85.71%	98.26%

Finally, to observe how the increase of the dataset through the GAN architecture improves the performance of the classifier, we trained the proposed classifier based only on the 20 reference genes (70% for training and 30% for testing) and compared with the trained classifier with augmented data. Table 5 compares the performance of both classifiers.

Table 5. Recall score in the augmented data and in the unaugmented data

Reference genes	*Recall* score	
	Training data	Test data
Augmented	98.40%	85.71%
Unaugmented	85.71%	66.66%

After training the classifier with the data augmented and based on the 4168 unclassified genes in the dataset, we consider as possible candidates for RG the genes for which the $f(x)$ function (Equation (5)) is equal to 1. Note that this function takes values equal to 1 for data within the decision boundary (positive class).

In this approach, the classifier detected 807 possible candidates to RG, reducing the amount of reference genes candidates to be tested in the laboratory by 80.64%. Figure 6 shows the 2-PCA representation of the RG and candidate genes selected by the proposed classifier. In this representation, we can perceive that the proposed classifier generates candidates close to the RG as we expected, which indicates that the mapped space created by the classifier is being consistent with the training data.

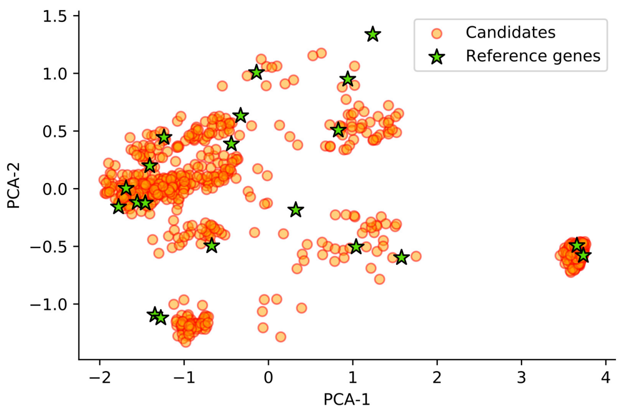


Fig. 6. Principal component analysis between the reference genes and the candidates.

Finally, based on the set of candidates for RG found in this approach and taking into account 27 candidates for RG found in [1]. We select the 11 candidates for RG that are common in both studies. Table 6 shows the common candidates in both studies.

Table 6. The most relevant candidates genes

Gene	Product
ftsX	Cell division protein ftsX
ftsY	Cell division protein ftsY
glyY	$\text{tRNA}_{\text{glyY}}$
mutY	Adenine DNA glycosylase
ndk	Nucleoside diphosphate kinase
nfuA	iron-sulfur cluster scaffold protein
rrsE	16S ribosomal RNA(rrsE)
rrsG	16S ribosomal RNA (rrsG)
spoT	(p)ppGpp synthase
thrW	$\text{tRNA}_{\text{thrW}}$
zupT	heavy metal divalent cation transporter ZupT

5 Conclusions and future works

This paper presented a new approach for *in-silico* identification of candidate reference genes from the *Escherichia coli* MG1655 dataset. This approach consisted of two main steps. First, a proposed GAN architecture was trained to synthetically augment the reference gene set. Second, with the augmented dataset was trained a one-class SVM classifier which obtained an 85% recall score in the test data. This new approach allowed the identification of 807 possible candidate genes for reference genes out of a total of 4170 expressed genes, which reduces the number of genes to be analyzed in the laboratory by 80%.

Finally, the approach demonstrated, as expected, that synthetically increasing the number of reference genes improves the classifier score, having in this research, an increase of 19% with respect to the trained classifier without the augmented data, making the selection of candidate genes more reliable.

For future work, other GAN architectures will be tested to try to improve the global optimum ($D(x) = \frac{1}{2}$). Thus, we could generate synthetic genes that are closer to the real distribution of the reference genes, being able to increase the score of the proposed novelty detector.

References

1. Berghoff, B.A., Karlsson, T., Källman, T., Wagner, E.G.H., Grabherr, M.G.: RNA-sequence data normalization through in silico prediction of reference genes: the bacterial response to DNA damage as case study. BioData mining, **10**(1), (2017)
2. Die, J.V., Román, B., Nadal, S., González-Verdejo, C.I.: Evaluation of candidate reference genes for expression studies in Pisumsativum under different experimental conditions. Planta, **232**(1), pp. 145–153, (2010)
3. Franco, E., Maués, D., Alves, R., Guimarães, L., Vasco, A., Arthur, S., Ghosh, P., Morais, J., Ramos, R.: A clustering approach to identify candidates to housekeeping genes based on RNA-seq data. Brazilian Symposium on Bioinformatics (in press), (2019)
4. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256, (2010)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In Advances in neural information processing systems pp. 2672–2680, (2014)
6. Kim, Y., YiSeul K., YH Kim.: Evaluation of reference genes for gene expression studies using quantitative real-time PCR in Drosophila melanogaster after chemical exposures. Journal of Asia-Pacific Entomology (2020)
7. KOZERA, B., RAPACZ, M.: Reference genes in real-time PCR. Journal of applied genetics, 391–406 (2013)
8. Legány, C., Juhász, S., Babos, A.: Cluster validity measurement techniques. In Proceedings of the 5th WSEAS international conference on artificial intelligence, knowledge engineering and data bases pp. 388–393. World Scientific and Engineering Academy and Society (WSEAS) Stevens Point (2006)

9. Ratle, F., Kanevski, M., Terrettaz-Zufferey, A.L., Esseiva, P., Ribaux, O.: A comparison of one-class classifiers for novelty detection in forensic case data. In International Conference on Intelligent Data Engineering and Automated Learning pp. 67–76. Springer, Berlin, Heidelberg, (2007)
10. Rocha, D. J., Santos, C. S., Pacheco, L. G.: Bacterial reference genes for gene expression studies by RT-qPCR: survey and analysis. Antonie Van Leeuwenhoek, **108**(3), pp. 685–693, (2015)
11. Sengupta, T., Bhushan, M., Wangikar, P. P.: A computational approach using ratio statistics for identifying housekeeping genes from cDNA microarray data. IEEE/ACM transactions on computational biology and bioinformatics, **12**(6), pp. 1457–1463, (2015)
12. Svingen, T., Letting, H., Hadrup, N., Hass, U., Vinggaard, A. M.: Selection of reference genes for quantitative RT-PCR (RT-qPCR) analysis of rat tissues under physiological and toxicological conditions. PeerJ, (2015)
13. Vandesompele, J., De Preter, K., Pattyn, F., Poppe, B., Van Roy, N., De Paepe, A., Speleman, F.: Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. Genome biology, **3**(7), (2002)
14. Wu, Q., Ma, X., Zhang, K., Feng, X.: Identification of reference genes for tissue-specific gene expression in Panax notoginseng using quantitative real-time PCR. Biotechnology Letters, **37**(1), pp. 197–204, (2014)
15. Krawczyk, B., Woźniak, M., Herrera, F.: On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. Pattern Recognition, **48**(12), (2015)
16. Pinto, A.C., de Sá, P.H.C.G., Ramos, R.T., Barbosa, S., Barbosa, H.P.M., Ribeiro, A.C., Miyoshi,A.: Differential transcriptional profile of Corynebacterium pseudotuberculosis in response to abiotic stresses. BMC genomics, **15**(1), (2014)
17. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. In Advances in neural information processing systems, pp. 582–588, (2000)
18. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems, **2**(1-3), pp. 37–52, (1987)