SORBONNE UNIVERSITÉ

CRÉATEURS DE FUTURS DEPUIS 1257

Master 2 BIM - BMC

Building accurate gene regulatory networks:

Network inference as a feature selection problem

Antoine Auvergne
Simon Chardin
Adrien Leroy

28^{th} November 2020

Abstract: In this project we review the GENIE3 algorithm, for gene regulatory network inference. We discuss its inner working, propose some modifications to the algorithm and compare its accuracy against other state-of-the-art methods on the DREAM4 in-sillico challenge. We find that GENIE3 is the most accurate method among its competitors for the specific network available during the DREAM4 challenge, and propose alternatives to the Random Forest and Extra Trees regressor used in its original implementation. The code used to generate the results presented here is available at the following URL: www.github.com/LeroyAdrien/GENIE3

Contents

1 Introduction

Since the early sixties, genes and proteins expression in the cell have proven to be regulated by the presence of a large set of molecules, among which, mRNA and proteins [1]. The expression landscape is therefore fundamentally self regulating, and being able to predict the influence of each gene expression on the expression of every other gene is mandatory to understand cell behaviour. The formalism of choice to describe the inter-regulation of genes is the Gene regulatory network (GRN) [2] in which each gene is reduced to a node and its influence to other genes as edges.

Inferring the correct GRN from expression data is therefore crucial to describe, and predict genes behaviour accurately. Various algorithms have been presented to tackle this challenge [3]. In this project we will describe the GENIE3 [4] approach and evaluate its performance against other protocols for GRN inferrence, mainly from noisy steady-state expression data generated for the DREAM4 (Dialogue on reverse-engineering assessment and methods) challenge [5].

2 Methods

2.1 Data used for the evaluation

The data used for the project has been taken from the DREAM4 In Silico Network Challenge. The goal of the DREAM4 computer network challenge is to reverse engineer gene regulatory networks from simulated steady-state and time series data. It challenged participants to infer the network structure from a given computer gene expression dataset. The network topology is obtained by extracting subnets from the transcriptional regulatory network of E. coli and Saccharomyces cerevisiae. They adjusted the subnet extraction method to give priority to including periodic network parts. The automatic adjustment interaction is deleted, that is, there is no self-interaction in the computer network. The file ***multifactorial.tsv*** contains steady-state levels of network changes, which are obtained by applying multifactorial perturbations to the original network. Each line gives the steady state of different perturbation experiments (i.e different changes in the network). For example, we can think of each experiment as a gene expression profile from a different patient. We are using the 5 multifactorial datasets with 100 genes.

Each dataset contains the expression values of each gene in several experimental conditions : raw expression, knockout,knockdown and dual-knockout. All of these data are concatenated in one matrix.

Thus, we have a sample of N measurements

$$LS = \{x_1, x_2, ..., x_N\} \tag{1}$$

where $x_k \in \mathbb{R}^p, k = 1, ..., N$ is a vector containing the expression values of p genes in the condition k :

$$x_k = \{x_k^1, x_k^2, ..., x_k^p\} \tag{2}$$

The goal is to make a prediction of the underlying links between genes. To do so, the algorithm assigns weights $w_{i,j} \geq 0 (i, j = 0, ..., p)$ to supposed regulatory links from any gene i to any gene j. The biggest values should correspond to actual regulatory interactions.

2.2 GENIE3 algorithm

Every step described here is taken from the original GENIE3 algorithm presented by Vân Anh Huynh-Thu et al[4].

GENIE3 generates a directed graph with p nodes where p is the number of genes. Each node represents a gene and an edge from node i to node j represents a direct regulation from gene i to gene j. The directed graph is created by decomposing the network between p genes in p regression problems. Each regression problem i represents the direct regulation of the gene i by all the nodes in the graph except i itself. This decomposition is based on the assumption that the expression of a gene is the function of the expression of all the other genes except itself. For each regression problem, every experiment in the dataset is considered.

For the k^{th} experiment, and with x_k^{-j} the vector containing all the expressions values in the kth experiment, the problem is formalized in this manner:

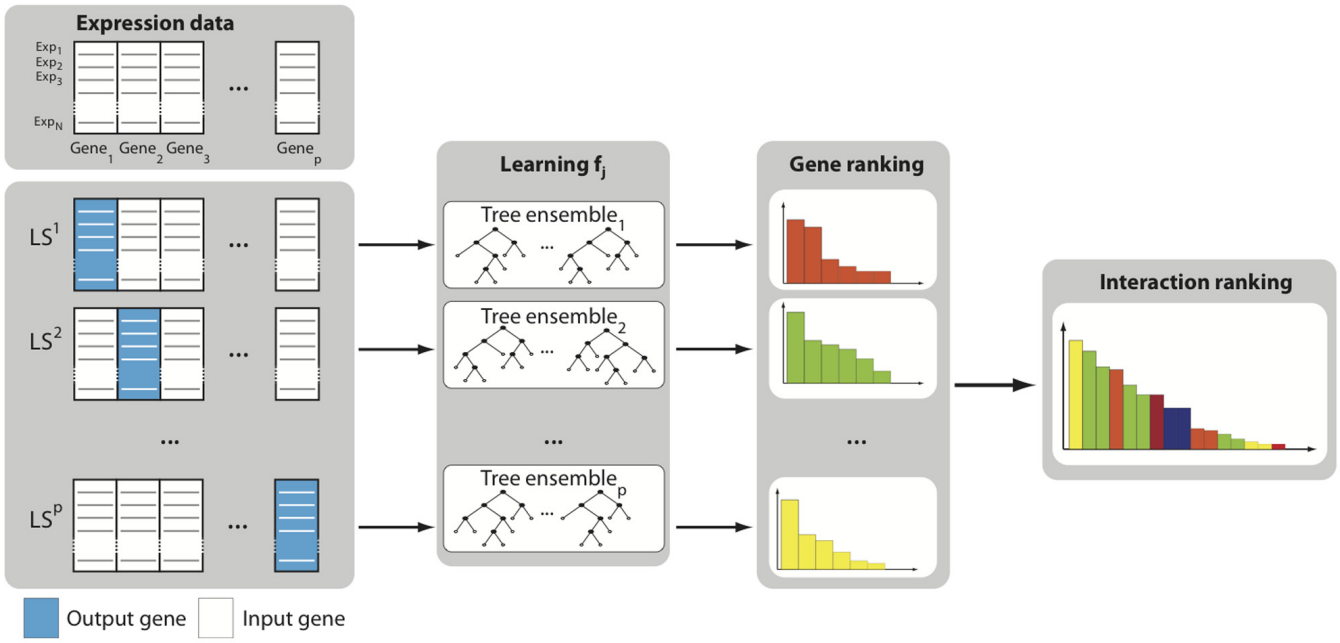

Figure 1: GENIE3 procedure

$$x_k^{-j} = x_k^1, ..., x_k^{j-1}, x_k^{j+1}, ..., x_k^{p^T} \quad (3)$$

$$x_k^j = f(X_k^{-j}) + \epsilon_k, \forall k \quad (4)$$

The goal of a tree-based algorithm is to learn a model in the form of a decision tree or a collection of decision trees that can predict the value of an output variable given the value of some input variable. Tree-based methods have been widely used to solve various problems in computational biology, such as DNA sequence annotation or biomarker discovery.

2.2.1 Bagging

In the Bagging (used in the "Bootstrap AGGregating" algorithm), each tree integrated is constructed from a copy of bootstrap, that is, a set of samples obtained through N random sampling, and replaces the original learning samples. Therefore, the selection of variables and thresholds on each test node is implicitly randomized by guided sampling. [6]

2.2.2 Random Forest method

Compared with bagging, this method adds an extra level of randomness. In the random forest set, each tree is constructed from the boot samples of the original learning sample, and at each test node, before determining the best split, K randomly selected (without replacement) from all input variables variable. When K is set to the total number of input variables, the random forest algorithm is equivalent to Bagging. [7]

2.2.3 Extra Tree method

In the Extra-Trees (for "extreme random trees") method, each tree is constructed from the original learning sample, but at each test node, the best split is determined among K random splits. Each split is determined by randomly selecting an input variable (no replacement) and a threshold (uniform selection between the minimum and maximum values of the input variables in a local subset of the sample). [8]

2.2.4 AdaBoost method

The AdaBoost algorithm is used to solve the prediction problem. However, for regulatory network in-

ference problem, we are more interested in which genes can be used most accurately predict the value of the target gene expression, rather than the predicted level of target gene expression. The prediction accuracy is represented by the supervisory interaction score. To perform the calculations, they used the variable importance score of the augmented tree, which was trained to predict the expression of the target gene from candidate regulators. The variable importance score of a single regression tree is calculated by using the variable to split the training sample and the contribution of the variable to the reduction of variance [9]. We calculate the variable importance score (VIS) of gene G in a regression tree by the following equation :

$$VIS(G) = |S|\,Var(S) - \left|S_{left}\right| Var(S_{left}) - \left|S_{right}\right| Var(S_{right}) \quad (5)$$

By taking all genes in the expression data as target genes and obtaining the VIS of the target gene candidate regulators, we can obtain the regulatory interaction scores of all gene pairs.

2.2.5 Gradient Boosting method

Gradient enhancement has also been successfully used to infer gene regulatory networks from steady-state gene expression data. Gradient boosting is a machine learning method that builds a regression or classification model by adding together weak learners (usually shallow decision trees). The gradient boosting algorithm follows the gradient descent process, which is used to minimize the loss L of the estimator f by adding the residual fitting estimator h [10]. The loss function L used in GRN is based on the following square error :

$$L(f(x_i), Y_i) = \frac{(Y_i f(x_i))^2}{2} \quad (6)$$

For each gene in the data set, a set of candidate transcription factor (TF) expression values is used to train the tree-based regression model to predict its expression profile. From the best-predicted TF to Target gene. All regulatory associations are merged and ranked by importance to finalize the output of GRN.

Compared with Random Forest, the bias reduction effect of gradient boosting can use shallower decision trees. Besides, it uses an early stop compared with GENIE3, the total number of decision trees constructed by GRNBoost2 is reduced by more than 80%.

2.2.6 TIGRESS

TIGRESS expresses the GRN reasoning problem as a feature selection problem and solves it by combining the popular LARS feature selection method with stable selection. Without any parameter adjustment, it ranked among the top 3 GRN inference methods in the 2010 DREAM5 Challenge [11]. GENIE3 also works with ensembles, but it is different from TIGRESS in that it uses a non-linear tree-based method for feature selection, while TIGRESS uses LARS. Given the obvious complexity of the problem, the linear relationship between TG and TF assumed by TIGRESS is difficult to understand, which can explain the difference in performance.

Lasso regression is a popular regularized linear regression that includes L1 penalty points. This has the effect of reducing the coefficients of input variables that do not contribute much to the prediction task. Least angle regression or LARS for short provides another effective method to fit lasso regularized regression models that do not require any hyper-parameters.

The coefficients of the model are found through an optimization process that attempts to minimize the error of the sum of squares between the prediction ($\hat{y}$) and the expected target value (y).

$$Loss = \Sigma_{i=0}^{n}(y_i - \hat{y}_i)^2 \quad (7)$$

2.3 Bayesian Ridge Score

As we have seen, most genes in GRN are regulated by a few regulators, so dynamic Bayesian networks and spline regression can be used to detect nonlinear interactions between genes and improve the results of GRN inferences.

Using the expression matrix, the Bayesian selection methods compute the probability distribution of genes Y_t given its parents expressed as:

$$p\left(Y_t \mid Y_{t-1}\right) = \prod_{g=1}^{G} p\left(Y_{g,t} \mid Pa\left(Y_{g,t}\right)\right)$$

The hierarchical Bayesian model is then computed as followed:

$$
\begin{gathered}
Y \mid X, \beta, \sigma^2 \sim N_n\left(X\beta, \sigma^2 I_n\right) \\
\beta_g \mid \sigma^2, \tau_g^2 \sim \gamma N_{m_g}\left(0, \sigma^2 \tau_g^2 I_{m_g}\right) + \left(1-\gamma_g\right)\delta_0\left(\beta_g\right) \\
g = 1, 2, \ldots, G \\
\tau_g^2 \sim \text{Gamma}\left(\tfrac{m_g+1}{2}, \tfrac{\lambda^2}{2}\right) \\
g = 1, 2, \ldots, G \\
\sigma^2 \sim \text{ Inverse Gamma } (a, b) \\
\gamma_g \sim \text{ Bernoulli } (p) \\
p \sim \text{Beta}(c, d)
\end{gathered}
\tag{8}
$$

One inference method of the gene regulation network is to find the structure N of the Bayesian network, which can better explain the data. There are many ways to find the structure of the Bayesian network, such as maximizing the probability of observation data (maximum probability, ML) or the posterior probability (maximum posterior probability, MAP) of the structure N of the given observation data [12].

Then, we calculate the link value between all genes and create a knowledge matrix (M_k). If $M_k(i, j)$ is less than the threshold, then the cell (i,j) in the knowledge matrix will be zero, otherwise, this cell will be 1.

2.4 Parameters variations

The impact of the hyperparameters can be studied by making them vary.

The number of trees varies between 200 and 2000 with a step of 200. It has an impact on all 4 methods tested: Extra Tree, Random Forest, AdaBoost, and Gradient Boosting (ET, RF, AB, and GB).

K can take two values, 'all' and 'sqrt', being respectively the number of features and $\sqrt{number\,of\,features}$. It has an impact on ET, RF, and GB.

The learning rate can take 10 values, being between 10^-3 and 10^0. It has an impact on AB and GB.

3 Results

3.1 Scoring methods to assess accuracy

The gold standard comparison can evaluate the accuracy of prediction based on two metrics: the "area under the curve" (AUC) score of the Receiver Operating Curve (ROC, also called the true positive rate and false-positive rate) and the precision and recall rate (PR) curve. The second comparison is to evaluate the prediction based on the inherent value of the prediction and the ability to introduce a specific prediction compared with the set of all prediction edges.

$$
\begin{aligned}
tp_{rate} &\approx \frac{Positives\,correctly\,classified}{Total\,positives} \\
fp_{rate} &\approx \frac{Negatives\,incorrectly\,classified}{Total\,negatives}
\end{aligned}
$$

Each discrete classifier produces (fp_{rate}, tp_{rate}) pairs corresponding to a single point in the ROC space. Thus, the diagram allows us to compare classifiers (model and/or parameterization) and choose the classifier closest to 1. The diagonal $y = x$ represents the strategy of random guessing. For example, if the classifier randomly guesses a positive classification half the time, it can be expected that half of the correct classification is positive and half of the negative classification gets the point (0.5, 0.5) in the ROC space.

GENIE3 uses RandomForest classifiers [4], such as naive Bayes classifiers or neural networks, which naturally generate instance probabilities or scores, which indicate the extent to which genes participate in interactions. These values can be strict probabilities, in which case they follow the standard theorem of probability. Or they can be general, uncalibrated scores, in which case the only attribute retained is that higher scores indicate higher probability. Although the outputs may not be appropriate probabilities, we can call them all probabilistic classifiers. Such ranking or scoring classifiers can be used with thresholds to produce discrete (binary) classifiers. If the classifier output is higher than the threshold, the classifier will produce 1, otherwise, it will produce 0.

Many bioinformatics pieces of research have developed and evaluated classifiers. These classifiers

will be applied to severely imbalanced data sets in which negative numbers are far greater than positive numbers. This is why it is recommended to use precision-recall (PR) plots because it can provide the viewer with an accurate prediction of future classification. The PR graph shows the accuracy value of the corresponding sensitivity (call) value.

3.2 Implementing new methods into GENIE3

Method	Extra Tree	Random Forest	Adaboosting	Gradient Boosting
$Mean_{AUROC}$	0.766	0.762	0.732	0.716
$Mean_{AUPR}$	0.172	0.18	0.152	0.132

Table 1: *Mean of areas under the curve over the five networks*

We first tried to improve GENIE3 results by implementing new algorithms (AB and GB) and results can be seen in table 1. This table is a summary of figures 9 and 10 (cf. Annexes). You can have an example of these curves with figures 2 and 3. The mean of both ET and RF methods are better than that of AB and GB methods. It could be a trade-off if these methods are faster.
Indeed, the runtime of GB is smaller than the runtimes of RF or ET. It could be an acceptable trade-off for very large datasets, but for the kind of data used in dream4, the time saved is on the order of a few seconds which does not make a significant difference.
We can conclude that the implementation of new methods in the GENIE3 algorithm didn't lead to an improvement in the results.

3.3 Comparing GENIE3 results with different hyper parameters

Thereafter, we decided to vary some parameters one at a time to see if we were able to get a better **AUROC**/**AUPR**. The details concerning the variations of the parameters can be found in 2.4.
We kept AB and GB methods to see if a better optimization might lead to better results than the original algorithm.

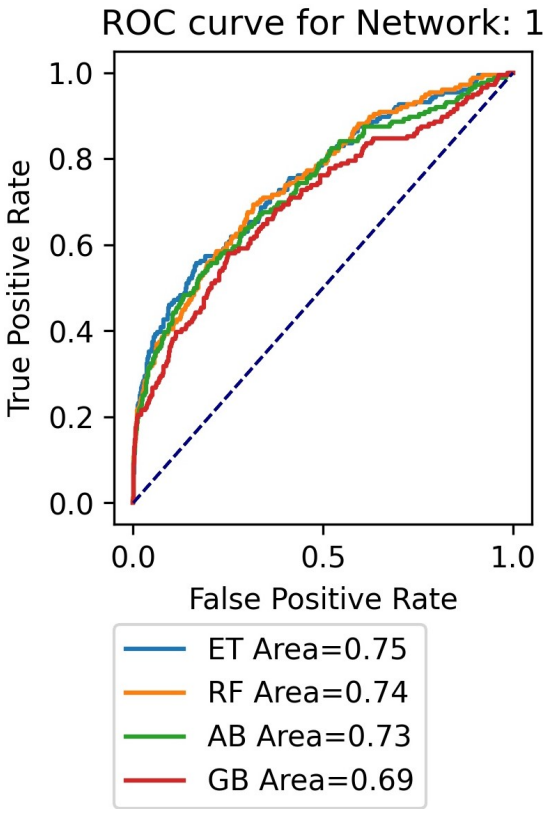

Figure 2: ROC curve for all methods on the first network

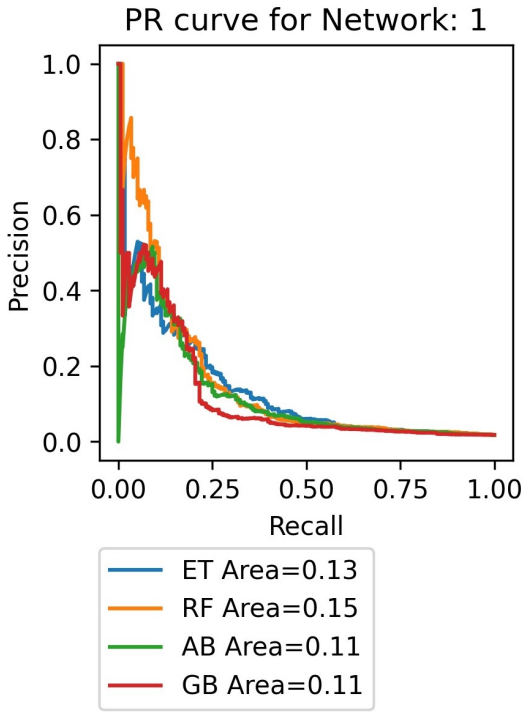

Figure 3: PR curve for all methods on the first network

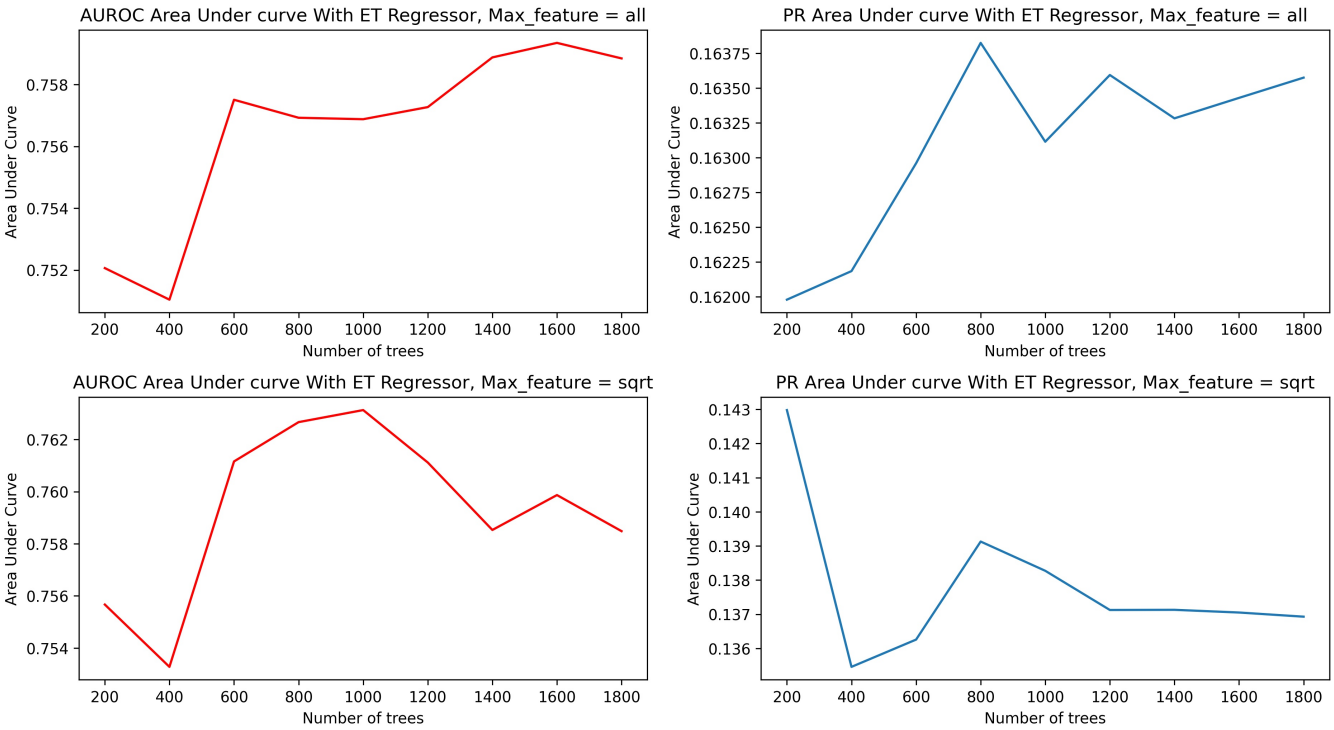

Figure 4: Optimisation of parameters for ET

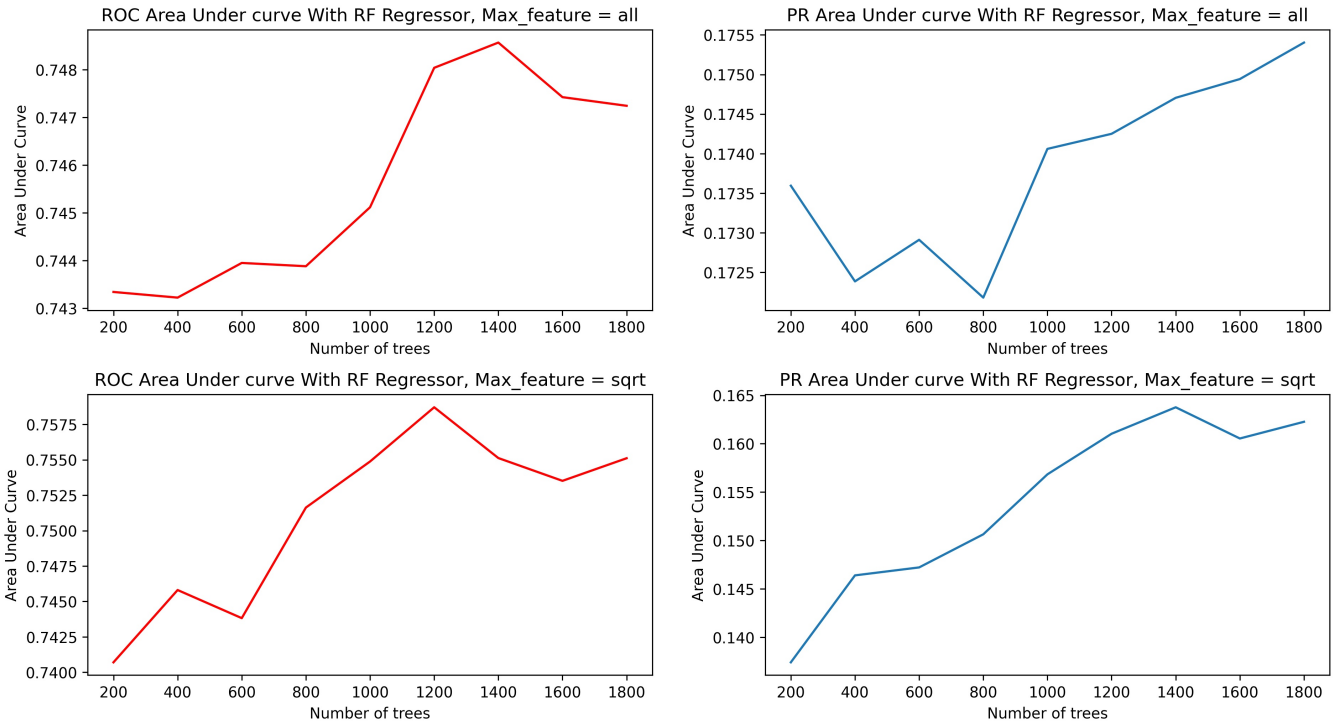

Figure 5: Optimisation of parameters for RF

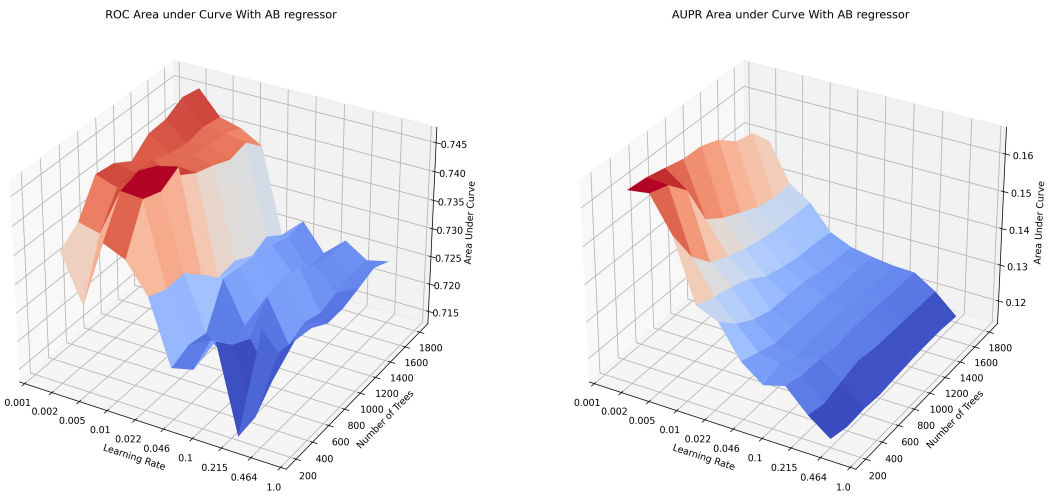

Figure 6: Optimisation of parameters for AB

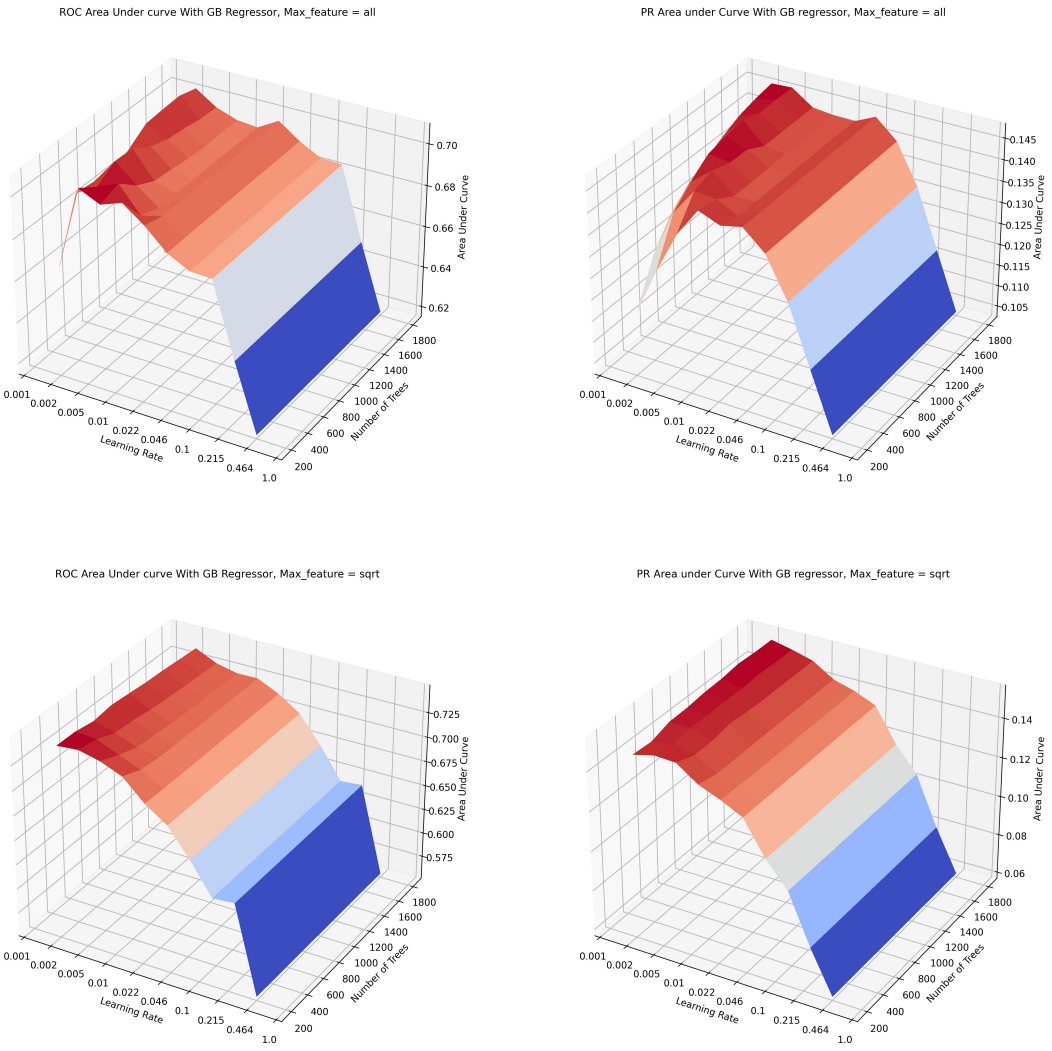

Figure 7: Optimisation of parameters for GB

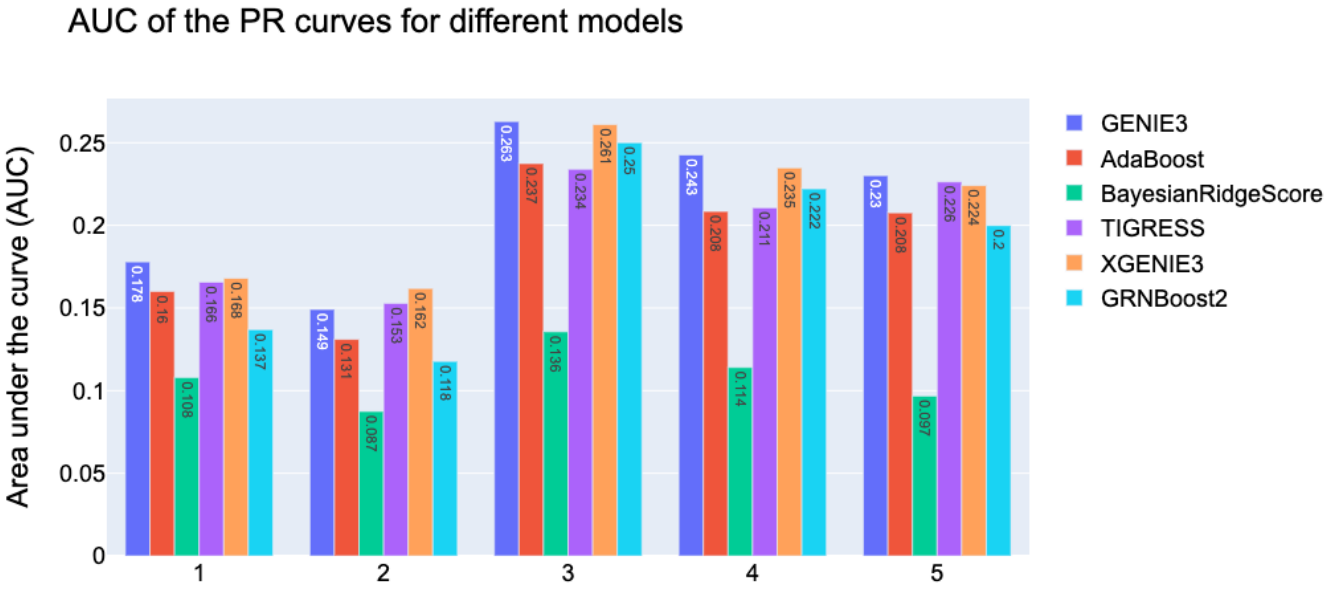

Figure 8: Comparing GENIE3 with other algorithms

We observe on figures **4** to **7** that the optimisation of the parameters admits many local minima. The resolution of this optimization, therefore, depends on the dataset we are studying and is not trivial (e.g. more features would equate to a better accuracy). Overall we can see that the optimization of the various parameter, does not allow us to get significantly different results when it comes to the AUROC and AUPR metrics, with an average value of respectively 0.754 and 0.165

3.4 Comparing GENIE3 with other inferring methods

Methods based on statistics and machine learning (ML) have recently made progress in the construction of gene regulatory networks (GRN) based on high-throughput biological data sets.

We compared a set of 6 feature selection methods based on two important criteria: accuracy and stability using the PR and ROC curves. **Figure 8** summarizes the relative performance of all methods and is worth mentioning.

Regarding the choice of method to train a classifier once features are selected, we observed that the best accuracy was achieved by GENIE3 or XGENIE3.

Six of the methods also need to specify one or more parameters. To this end, we separately estimated the parameters of each method of the synthetic network data set, the curated model data set and the real data set and provided them with the parameters that led to the best AUPRC value.

Regarding verification, many techniques have been adopted to evaluate the performance of network reasoning methods: simulation, enrichment analysis, document support, expert interpretation, and laboratory verification. In simulation studies, the ground truth (reference network) is known and used to assess the quality of the constructed network [13].

In the SCENIC pipeline, the focus is on the ability to process single-cell data from the acquisition to the results, and the focus is on using different correlation metrics to model the pairwise relationships between genes: empirical Bayes and information metrics use information theory techniques.

Single-cell data provides the expression of individual cells, many of which are from different developmental stages. This makes it possible to arrange

cells in a time-wise manner, which can simulate the expression changes of individual genes over time [14].

These metrics are designed to capture not only the information shared between two genes but also the influence of other genes. That is why it is of great use to continue and test different methods and parameters to find the best key to unlock the pattern behind the data mined. The application of these methods has not been proposed in many real-world studies. This may be due to technical limitations in the scRNA-seq platform or the heterogeneity of single-cell data. The verification and calculation difficulties of the output network also raise questions about its applicability.

4 Discussion

Although GENIE3 proved to be the most accurate GRN algorithm among all the state-of-the-art techniques presented in this project. The DREAM4 in sillico challenge dataset is the only one we used to make that statement. In its original presentation, GENIE3 was described as equally accurate on expression data from Escherichia. Coli but we were not able to validate this statement because of a lack of time and computational resources. Exploring the algorithm GRN inference from in-vivo data, against a curated dataset on model organisms, would have been the next step in our research. It is important to note that in DREAM4, two dataset sizes were given to train and test the model. The purpose of constructing GRNs is to deeply understand the regulation of genes and/or transcription factors in the process of cell state transition. Therefore, the network is usually specific to the cell type. However, most GRN inference methods produce a single network containing all possible interactions. This may be useful in small-scale experiments where only one or a few cell types and genes are involved. We decided to study the 100 gene network to have a good overview of the behavior of such models on a not-so-small set, to which the model would over-fit, but still keep the sense of significance to real life and the scale it represents.

On the topic of scale, it is clear that with the rapid development of sequencing technology, the number of cells available in each data set has increased exponentially. And as we saw trying to compute different dataset from E.Coli to some network studied from mice and also Humans (always shoot for the moon), the compute time on our humble home machines (but also on Google Colab), was not realistic for us to have a good sense of any result produced. This is a Bottleneck that those algorithms have to face and even if GRNBoost2 tries to fix those downfalls by not exploring all the trees to their end, the size of the data is still becoming too quickly too much to handle without a specialized cluster. Besides, unlike other analyses that can use dimensionality reduction techniques to reduce complexity, the GRNs method needs to analyze all features/genes to provide a comprehensive gene network.

5 Conclusion

In this article, we review GENIE3 and compared it with other methods to improve the accuracy of GRN reconstruction from gene expression data. The results from the DREAM4 challenge show that the proposed method is significantly better than other state-of-the-art methods for this specific challenge. Through this review, our main objective was to further our knowledge in GRN, and algorithms applied in this field but also to try to create a small panel of benchmark of similar statistical models to present how each of them is handling those specific data, and how their features performs. To accomplish these objectives, we first took a good look under the hood of GENIE3. We then discuss in-depth the hypothesis and the techniques this method uses to model the GRN. We then divide different parameters into the methods to observe the impact they have on the results, and thus further deepening the understanding of the interworking of the Decision Tree statistical process. We also compared GENIE3 to other different machine learning algorithms, which ended up proving the worth of GENIE3 on the dataset of DREAM4. We finally discuss the challenges that future methods need to address to generate more accurate and comprehensive GRN from single-cell data and so based on the difficulties we faced and hardships we witness during this exploration.

6 Bibliography

Articles

[1] François Jacob and Jacques Monod. “Genetic regulatory mechanisms in the synthesis of proteins”. In: *Journal of Molecular Biology* 3.3 (1961), pp. 318–356. ISSN: 0022-2836. DOI: https://doi.org/10.1016/S0022-2836(61)80072-7. URL: http://www.sciencedirect.com/science/article/pii/S0022283661800727.

[2] E. Davidson and M. Levin. “Gene regulatory networks”. In: *Proc Natl Acad Sci U S A* 102.14 (Apr. 2005), p. 4935.

[4] Vân Anh Huynh-Thu et al. “Inferring Regulatory Networks from Expression Data Using Tree-Based Methods”. In: *PLOS ONE* 5.9 (Sept. 2010), pp. 1–10. DOI: 10.1371/journal.pone.0012776. URL: https://doi.org/10.1371/journal.pone.0012776.

[5] G. Stolovitzky, D. Monroe, and A. Califano. “Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference”. In: *Ann N Y Acad Sci* 1115 (Dec. 2007), pp. 1–22.

[6] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.

[7] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[8] Pierre Geurts, Damien Ernst, and Louis Wehenkel. “Extremely randomized trees”. In: *Machine learning* 63.1 (2006), pp. 3–42.

[10] Sungjoon Park et al. “BTNET: boosted tree based gene regulatory network inference algorithm using time-course measurement data”. In: *BMC systems biology* 12.2 (2018), pp. 69–77.

[11] Anne-Claire Haury et al. “TIGRESS: trustful inference of gene regulation using stability selection”. In: *BMC systems biology* 6.1 (2012), p. 145.

[12] Matthieu Vignes et al. “Gene regulatory network reconstruction using Bayesian networks, the Dantzig Selector, the Lasso and their meta-analysis”. In: *PloS one* 6.12 (2011), e29165.

[13] Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. “The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures”. In: *PloS one* 6.12 (2011), e28210.

[14] Hung Nguyen et al. “A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data”. In: *Briefings in Bioinformatics* (2020).

Books

[3] Vân Anh Huynh-Thu Guido Sanguinetti, ed. *Gene Regulatory Networks, Methods and Protocols*. 2019. DOI: 10.1007/978-1-4939-8882-2. URL: https://app.dimensions.ai/details/publication/pub.1110579093%20and%20http://repositories.lib.utexas.edu/bitstream/handle/2152/6533/bhingea46536.pdf.

[9] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.

7 Annexes

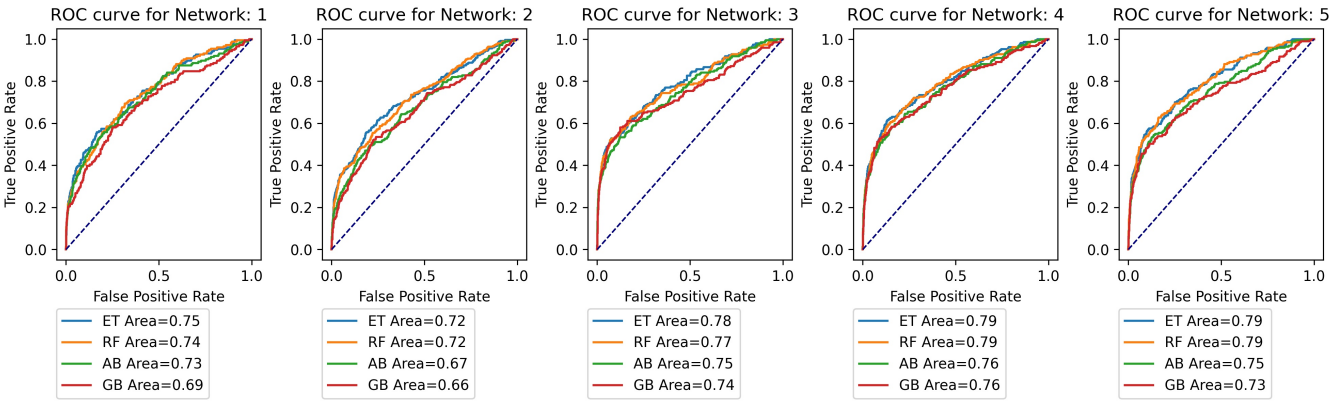

Figure 9: Comparison of ROC curves for differents implementations in GENIE3

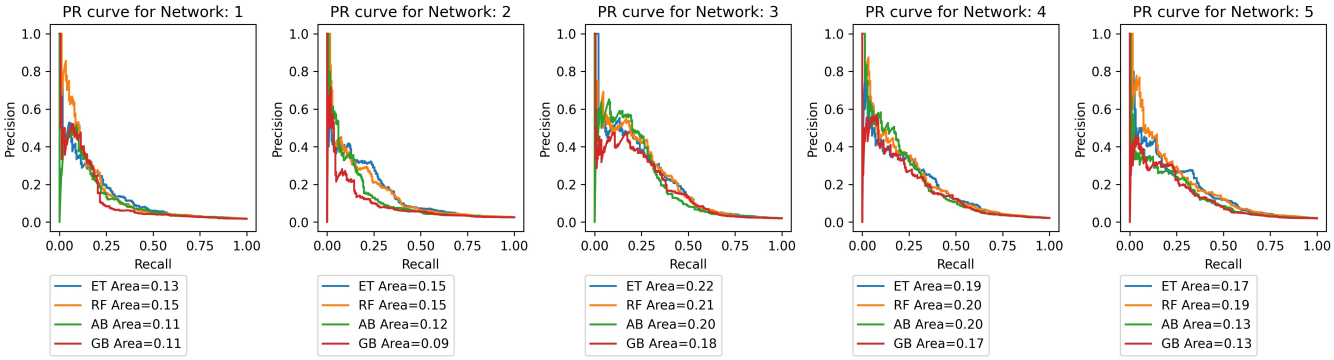

Figure 10: Comparison of PR curves for differents implementations in GENIE3