University of Liège - Faculty of Applied Sciences

# Local machine learning-based feature importances for gene regulatory network inference

Final work carried out with the aim of obtaining the degree of Master "Computer Engineering” by

**Alexandre KERFF**

*Supervisor* :

Prof. Pierre GEURTS

Academic year 2023-2024

# Acknowledgement

My heartfelt appreciation goes out to everyone who helped make this work a reality, whether directly or indirectly. Your help has been really helpful, and I sincerely appreciate it.

I am particularly grateful to my supervisors Pierre Geurts and Vân Anh Huynh-Thu for their guidance, support and disponibility. They have frequently given me smart suggestions to improve my work, responded to all of my inquiries, and given me wise counsel.

I would also like to express my sincere thanks to my friends for their invaluable help in completing this thesis. Their emotional and logistical support, the many hours they spent re-reading my work, as well as the many hours they devoted to my research, all contributed to the realisation of this thesis.

I express my gratitude to all the inviduals cited above.

# Abstract

Understanding how a cell (or organism) reacts to a change in the environment or disturbance requires an understanding of the intricate processes controlling gene expression and, therefore, protein synthesis. A common representation of these mechanisms is the gene regulatory network, that aims at defining the regulation links between genes as a set of interactions. Inferring those gene regulatory networks from expression data has been a widely studied field at the level of bulk expression data. However, recent breakthroughs in sequencing technologies enables measurements at the resolution of a single cell. Such data allows the development of research towards the analysis of gene regulatory networks for a single specific cell or for a distinct cell type, rather than global interactions. This thesis has the objective to perform these analyses.

Exploiting the foundations of a technique elaborated for bulk data, `Genie3` [1], the problem of cell-specific and cell-type specific network inference can be addressed by the means of local feature importance methods instead of global algorithms.

To this extent, this study first examines numerous local feature importance methods and provides new implementations for a few of them. It evaluates them with respect to the global methods on simple regression problems. Analysing these techniques highlights particular methods of interest with promising results (`Shap`, Saabas, local mean decrease of impurity, and local mean decrease of accuracy).

Subsequently, the local methods are employed to address the cell-specific network inference issue in order to examine their applicability in this domain. To evaluate the anticipated local networks' capacity to identify distinct interactions, they are contrasted with global networks on a synthetic dataset. It is demonstrated that analysing the local feature importance algorithms (`Shap`, Saabas, local mean decrease of impurity, and local mean decrease of accuracy) yields more accurate findings at single-cell resolution than analysing the global networks.

Next, the efficient local methods are investigated in the context of cell-type specific network inference. This problem is addressed in the thesis by averaging the local scores of cells sharing the same types on a synthetic dataset with mixed types. Comparing the methodology to the application of global method to each separated type, it is shown that all the local algorithms

perform poorly.

Subsequently, a real dataset containing gene expression data from several cell types obtained from peripheral blood is subjected to local techniques. As there are no ground truth networks accessible, the cell-type feature importance inference is performed to recover rankings of features scores. Common and unique interactions between types are highlighted by comparing the most important significance values for each type. The use of local mean decrease of impurity is shown to identify common and different rankings than global methods run on the separated datasets.

Lastly, certain genetic markers found in each patient that contributes to the actual dataset are examined. A correlation is calculated between networks created from patients who have the genetic marker and patients who do not. A few low correlation values found by local mean decrease of impurity enable the identification of certain indicators that affect gene regulation.

# Contents

# List of Figures

# Introduction

Fundamental biology study aims increasingly to understand the intricate systems that control how a cell responds to changes in its environment. To this extent, since gene regulation is the primary mechanism defining a cell's response to perturbations, variables influencing gene regulation are of great relevance. A common representation of gene regulation is the representation using networks defining links of inhibition or activation between different genes. This representation is called gene regulatory networks (GRN). The inference of GRNs from experimental data is a major challenge in computational biology and a widely studied field. Many approaches use machine learning to solve this problem, such as `Genie3`, a framework developed to infer GRNs using ensemble of regression trees from bulk gene expression data.

The utilization of bulk expression data — that is, gene expression measurements at the level of the body, organs, or tissues — has been the main focus of previous GRN inference research. Recent advances in technology have made it possible to detect gene expression at the single-cell level. The inference of regulatory interactions unique to a particular cell type, or cell-specific regulatory network inference, is theoretically made feasible by such data. Such discoveries would allow an in-depth understanding of complex and unique regulation mechanisms. But to solve this issue, new inference techniques must be created. Although a number of network inference techniques for single-cell data have been put forth in the literature, the issue is still difficult, and these techniques have not yet matured to the same extent as GRN inference techniques using bulk expression data.

This thesis aims to address these gaps using methodology that has not been studied in the literature. A machine learning regression model is trained from a dataset of single-cell measurements to predict the expression of a given gene from the expressions of all other genes (or a subset of candidate transcription factors). A cell-specific network can then be derived from local importance scores derived from the machine learning model for each cell (or cell type) in the dataset. This approach follows the regression approach proposed in GENIE3, but instead of using global scores, it uses local measures to solve the problem of cell-specific network inference. This is a simple extension of GENIE3, although it hasn't been thoroughly examined in the literature.

The assessment and comparison of several local significance scores for cell-specific gene network inference from single-cell data is thus the aim of this master's thesis. Several methods that may be used to obtain local significance scores from various machine learning models exist, and this thesis will mainly focus the derivation of local scores from regression tree ensembles.

First, in Chapter 1, this paper begins with a comprehensive introduction to the concepts of machine learning models and metrics. It explores the foundation of machine-learning related tasks and introduces tree-based models, the basis of the procedure explored in this thesis. An explanation of gene regulation fundamentals is also provided, and a description of the GRNs, single-cell GRNS, and cell-type GRNs inference methodology is presented.

Then, in Chapter 2, an in-depth description of global and local feature importance methods is introduced. As the detection of the regulation links is brought by the use of local feature scoring techniques, these techniques constitute the main work of this thesis. Some of the explored techniques are existing implementations (`Shap`, Saabas, local mean decrease of impurity, fair-equivalent-symmetric-perturbations, and equal-surplus value) while others are implementations and derivations of existing values (local mean decrease of accuracy implementations). A review of normalisation techniques that will be used is also provided.

Next, in Chapter 3, datasets explored in this thesis are presented. The methods will be confronted first to a simple regression problem, Friedman 1, to verify their working. The synthetic generation of datasets for single-cell GRN inference with Dyngen is reviewed, as well as the methodology chosen in this thesis to mimic cell-type representations and generate corresponding datasets with boolODE. A real dataset issued from peripheral blood data is also introduced.

Finally, in Chapter 4, the results obtained using the local methods on each dataset together with the corresponding methodology to obtain these results are presented. A first evaluation considers the mean area under the receiver operating characteristic curve (AUROC) and area under the precision recall curve (AUPR) on the Friedman1 dataset. Spearman's correlation is also explored on this dataset to measure similarity between methods. The assesment of the AUROC and AUPR measures is then performed on single-cell and cell-type GRN inference synthetic datasets. The impact of normalisation is also addressed. Finally, the real dataset is explored, comparing the feature rankings of one local method (localMDI) and global methods (Genie3). Another analysis is also carried out by comparing the impact of genetic markers on the feature importance scores, by computing the Spearman's correlation between the local feature importance scores of patients with and without the markers.

# Chapter 1

# Background

In the context of gene regulatory network inference, it is important to understand the context of machine learning models and gene regulation. In particular, the foundation of the methods used in this thesis are regression tree-based models. To better understand the task of the thesis, a short analysis of what is gene regulation and gene regulatory network inference is as well mandatory.

This chapter will describe all the steps required to understand what are random forest regressors, from the basic principles of machine learning, and also develop the concept of gene, gene regulation networks and how to infer these.

## 1.1 Machine Learning models

In this section, a closer look about what is machine learning will be provided. Especially, highlight will be made on the regression task. Since this one is here solved with tree-based model, explanations about what is a regression tree and in particular what is a random forest regressor will be given.

### 1.1.1 What is Machine Learning ?

*"Machine learning is concerned with the design, the analysis, and the application of algorithms to extract a model of a system from the sole observation (or the simulation) of this system in some situations (i.e., by collecting data)"*??.

The goal of machine learning is thus to define a **model** through algorithms applied to a system. This model can describe relationships and, more generally, provides information about the system it is based on. It can even be used to make predictions about unseen data.

In machine learning, problems of interest can be divided in four classes: supervised learning, unsupervised learning, semisupervised learning and, reinforcement learning. The problem studied in this thesis concerns supervised learning.

Formally, *"From a learning sample $\{(x_i, y_i)|i = 1, ..., N\}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, find a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the expectation of some loss function $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ over the joint distribution of input/output pairs"* [2].

$$E_{X,y} l(f(x), y) \tag{1.1}$$

The learning sample, from which the model will be built, thus contains data (features $x_i$) and their corresponding labels $y_i$. The model $f$, will be built learning from these data and labels. Regression, in particular, is a problem of machine learning where the outputs and labels are quantitative values. The most common loss function in regression is the *squared error loss* [3]

$$L(Y, f(X)) = (Y - f(X))^2 \tag{1.2}$$

Supervised learning has two main goals : either make predictions on unseen data, either studying relationships between features and outputs, which in this thesis will be the principal objective [2].

### 1.1.2 Regression tree

Tree-based methods' goal is to divide feature space into different regions. In particular, it is based on binary trees partitioning (figure 1.1) [3].
The algorithm (CART) used to build the model is the following:

1. Choose a pair of feature $k$ and threshold $t_k$ that achieves the best split of the feature space in 2 regions (figure 1.1)

2. If needed, restart splitting on and on subsets, recursively.

3. If a criterion is reached, the node is a leaf of the tree (end node) and the value of the output is attached to it

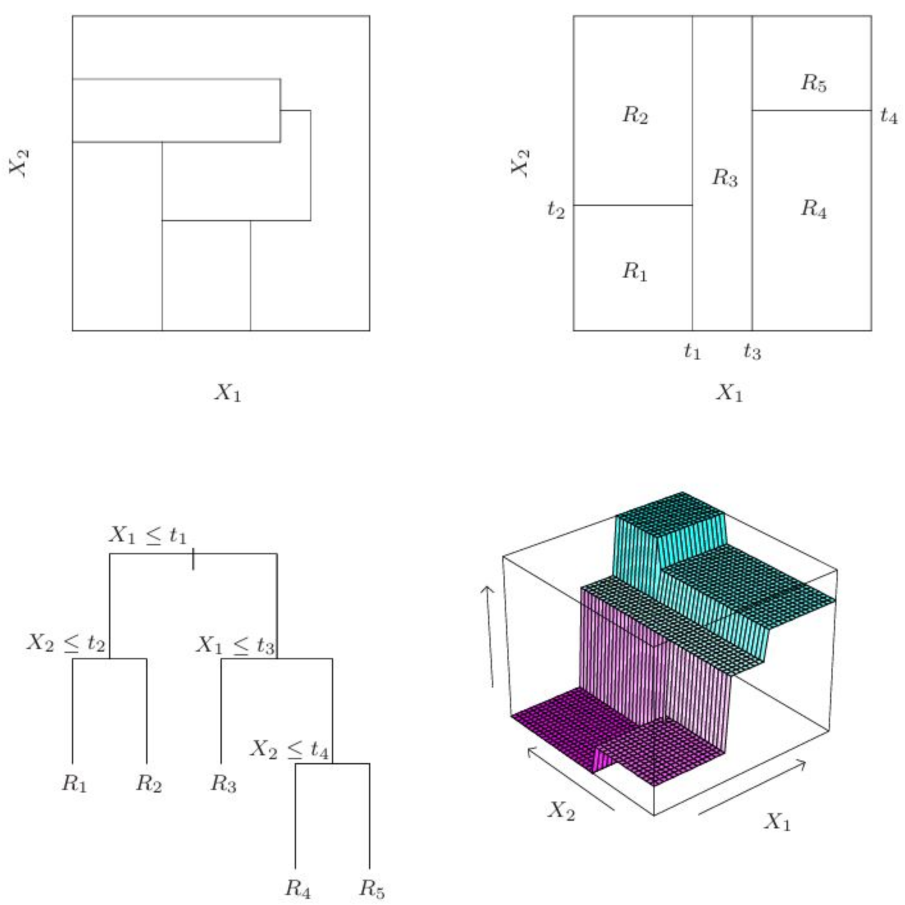


Figure 1.1: *Regression Tree growing through CART algorithm example. Visualization of the feature space division leading to a tree structure [3].*

The best possible split is defined by the split that maximize the reduction of impurity $i(t)$. This measure of impurity can be defined by several different ways. In the case of regression trees, the impurity measure is based on the mean squared error loss.
The split can then be determined by looking for the feature that causes the biggest reduction of variance in the child nodes [4].

The function the algorithm tries to minimize for each pair split $k$ with threshold $t_k$ is the following :

$$J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right} \tag{1.3}$$

with $MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^{(i)})^2$ and $\hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)}$ and $m_{left/right}$ representing the amount of instances of the learning sample in the corresponding subset [5].

As far as the criterion for stopping the tree's growth is concerned, many approaches can be considered, conducting to different degrees of overfitting. Overfitting occurs when the model matches too precisely the learning set, leading to poor predictions on unseen data. The basic option chosen in this case was to grow the tree until there are fewer samples left in the node to split than a specific value `min_samples_split`.

The representation of a tree is thus a "tree" of nodes, testing each a particular feature. Depending on the threshold, each feature will either follow a path to the right or the left node, where another test on a feature will be conducted. It will eventually reach a terminal ("leaf") node, where an output value will be given depending on the leaf reached.

This structure is great for its interpretability. Indeed, it is straightforward to follow a path of the tree and understand which splits and features contributed to a prediction.

### 1.1.3 Random Forest Regressor

One main default of regression and decision trees is that they have a high variance and instability. Indeed, a small perturbation in the data is sufficient for one splitting node and often all subsequent nodes to change, leading to a totally different tree. This instability is hard to avoid in the model, even trying to modify splitting or stopping criteria. One possible solution is through bagging [3].

Bagging is a method for decreasing an estimated prediction function's variance [3]. The main mechanism behind bagging is just to average many models to reduce the variance. Random Forest is thus a model fitting many regression trees and averaging their predictions (figure 1.2). This model was chosen in this thesis due to its performance while keeping a high degree of interpretability, allowing some specific methods of feature importance to work.

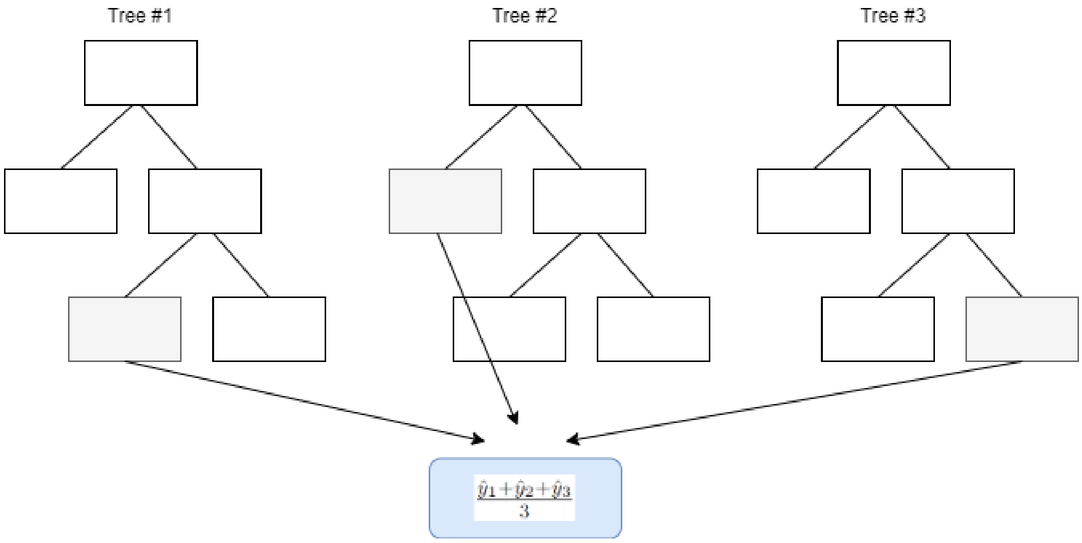


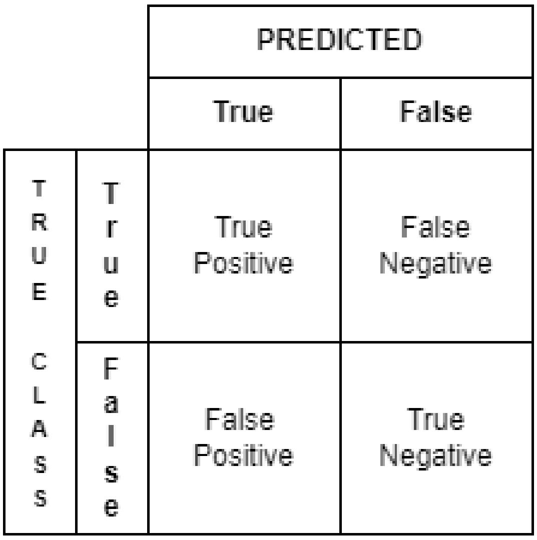
Figure 1.2: *Random Forest schematic. Predictions are aggregated.*

## 1.2 Measurement tools

In this section, scoring tools and ways to assess the efficiency of the methods will be presented. The area under the receiver operating characteristic curve (AUROC) and the area under the precision recall curve (AUPR) are common tools to evaluate the quality of classification methods. Both are used in this thesis to quantify the quality of classification of existing and absent regulation links between genes of the methods used. Another metric, Spearman's rank correlation, is a tool to evaluate correlation between values. It is applied to the results of the different methods used in this thesis to compare their similarities.

### 1.2.1 AUROC & AUPR

In the context of a classification problem, a confusion matrix can be defined to extract four useful metrics [6] (figure 1.3).



|  |  | PREDICTED |  |
|---|---|---|---|
|  |  | True | False |
| TRUE CLASS | True | True Positive | False Negative |
|  | False | False Positive | True Negative |

Figure 1.3: *Confusion Matrix. Metrics are defined, comparing values predicted by the model and ground truth values.*

1. **Precision** represents the proportion of accurate True predictions, i.e.
$$precision = \frac{TP}{TP + FP} \tag{1.4}$$
2. **Recall** defines the proportion of positive instances determined as true by the classifier, i.e.
$$recall = \frac{TP}{TP + FN} \tag{1.5}$$
3. **False positive rate** (FPR) is the proportion of predicted true on all negative class, i.e.
$$FPR = \frac{FP}{FP + TN} \tag{1.6}$$

Two particular curves can be deduced from these metrics: receiver operating curve (ROC) and precision-recall curve (PR). The first one represents the recall as a function of false positive rate. Each classification model prediction is attached to a threshold influencing its results, and the ROC curve represents the recall/FPR for different values of threshold.
Similarly, the PR curve can be defined as the curve defining the precision related to the recall for different thresholds. This specific metric defines a trade-off. Indeed, the more the threshold is decreased to have more positive instances well classified and a good recall, the more the risk to have false positives and, to lower the precision increases. [6]

To study the performance of a classifier, a common way is thus to determine the

area under (AU-) these two curves to get rid of a specific threshold. Area under the receiver operating characteristic curve (AUROC) is a score that determines the ability of a model to distinguish classes, with the best score being 1, while area under the precision-recall curve (AUPR) is an indicator of how precise the classifier is, the best score being once again [6].
In this thesis, even if the problem is about regression, the feature importance values will be considered as being scores used to determine existing (True) or not (False) regulatory links. Thresholds will then be used on determined scores to deduce the curves, and thus the AUPR/AUROC.

### 1.2.2 Spearman's rank correlation

Spearman's rank correlation $r_S$ is a measure of correlation between vectors of values. Considering a ranking of the observations of 2 samples $u$ and $v$ of size $n$, and the corresponding rank of their $i^th$ observations $u_i$ and $v_i$, one can define Spearman's rank correlation $r_S$ as (1.8).

$$
\begin{aligned}
r_S &= \frac{n\sum_{i=1}^{n} u_i v_i - (\sum_{i=1}^{n} u_i)(\sum_{i=1}^{n} v_i)}{\sqrt{(n\sum_{i=1}^{n} u_i^2 - (\sum_{i=1}^{n} u_i)^2)(n\sum_{i=1}^{n} v_i^2 - (\sum_{i=1}^{n} v_i)^2)}} \qquad (1.7) \\
&= 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \text{ where } d_i = u_i - v_i \qquad (1.8)
\end{aligned}
$$

With equation 1.8, the differences $d_i$ are a good approximation if those differences are not too small [7].
In this thesis, Spearman's rank correlation was computed using `scipy` library and corresponding function `spearmanr`. This correlation value was computed to compare the rankings of the methods on a same problem.

## 1.3 Gene Network Inference problem

This thesis aims to apply feature importance methods from machine learning to infer gene regulatory networks, especially cell-types and cell-specific regulatory network. To effectively address this task, it is essential to first explore the fundamental concepts underlying gene regulatory networks.

The following section provides a comprehensive overview, beginning with the definition and the structure of a gene, followed by the mechanisms of gene regulation,

and culminating in the concept of gene regulatory network inference.
The section concludes with a discussion on the methodology of gene network inference derived from the existing algorithm `Genie3`.

### 1.3.1 Genes and Gene Regulation

#### Definitions

A gene is a specific region of DNA that encodes information responsible for particular characteristics of an individual. A gene consists of sequences of coding nucleotides (exon) interspersed with non-coding sequences (intron).
In particular, DNA is a double-helical structure comprising different nucleotides, and is present in all cells. It contains the genetic material of an individual.

Nucleotides are organic molecules distinguished by their specific base. There are only a few possible bases, and the uniqueness of DNA is determined by the sequence of these nucleotides. RNA, or ribonucleic acid, is transcribed from a single strand of DNA.

#### Protein synthesis

Proteins, and the process of their synthesis, form the link between genes and phenotypic traits-the observable characteristics of an individual. Gene expression is the process by which the information in genes directs proteins synthesis, comprising two main steps: transcription, and translation.

During transcription, an enzyme called RNA polymerase separates the DNA strands in two. Only one of the two strands is transcribed to synthesise a complementary RNA strand. In eukaryotic cells, which have a defined nucleus, transcription initiation is regulated by transcription factors that bind to a specific region of the DNA known as the promoter. The RNA strand produced during transcription is known as messenger RNA (mRNA), which serves as the template for translation — the process by which proteins are synthesised based on the sequence encoded in the mRNA (figure 1.4)[8].

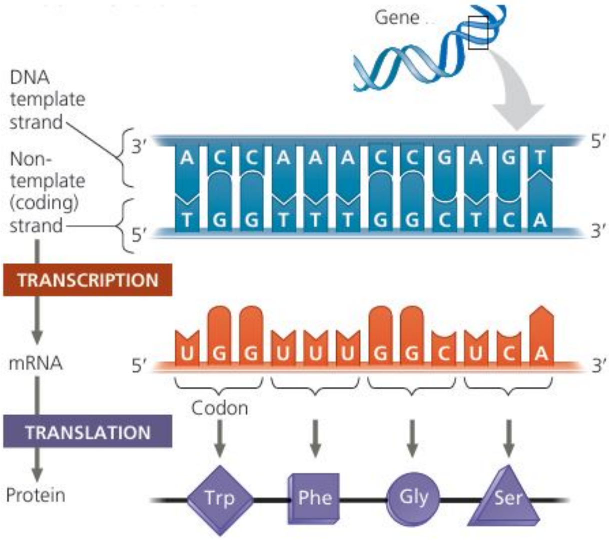


Figure 1.4: *Gene expression steps. Gene is translated in messenger RNA during transcription and then translated in a protein. [8].*

### Gene regulation

Cells must regulate gene expression in response to environmental changes. While gene expression is often regulated at the transcription level, it can also occur at other times (after the transcription, chromatin modification, ...).
Despite containing the same genome, different cell types express distinct subset of genes, which allows for cellular differentiation and specialisation. This regulation is crucial for adaptation and is mediated by transcription factors that interact with specific regions of the DNA, known as *enhancers* and *promoters*, see figure 1.5. The speed and efficiency of gene expression are influenced by these interactions, with activators and repressors binding to these control elements to modulate transcription rates.

Importantly, gene regulation is often non-linear, with the combination of control elements in an enhancer region being more critical than the presence of a single control element [8].

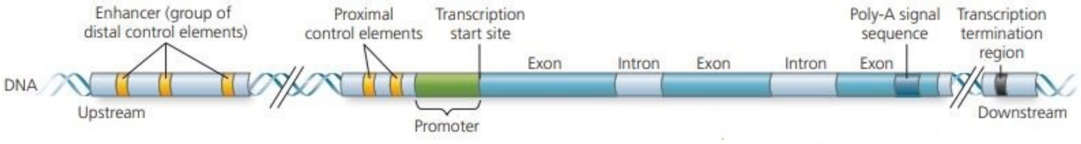

Enhancer (group of distal control elements)
Proximal control elements
Transcription start site
Poly-A signal sequence
Transcription termination region
DNA
Upstream
Exon
Intron
Exon
Intron
Exon
Promoter
Downstream

Figure 1.5: *Gene structure. Enhancers and control elements are key targets of gene regulation. The gene consists of coding regions (exon) and non-coding regions (intron), and includes start and stop signals for transcription [8].*

In a cell, activators and repressors production is itself regulated by the expression of specific genes, leading to complex structures and sequences of differential gene expression, a cascade of specific interactions between the speed of expression of a subset of genes. These intricate sequences of interactions are referred to as **gene regulatory networks**.

### 1.3.2 Gene Regulatory Networks inference & Genie3

The gene regulatory network inference problem involves predicting interactions from regulator genes $r$ to their target genes $t$ within a given gene subset. Numerous methods have been developed and bench-marked for inferring global gene regulatory networks across a set of genes and a sample of cells [9]. A key method of interest in this thesis is the `Genie3` algorithm [1].

`Genie3` decomposes the problem of the gene regulatory networks inference for a set of $p$ genes into $p$ regression problems. Each subproblem involves predicting the expression level of one gene $p$ from the expression levels of all other genes in the dataset (figure 1.6).
For each subproblem, a random forest model is trained on the gene expression data, followed by the application of a global feature importance algorithm is then run on the trained model. This algorithm assigns an importance score to each feature, reflecting the significance of the feature to predict the output.

A high importance score for a feature $f$ indicates that the gene $f$ is likely to regulate the output gene $p$. From this hypothesis, it is possible to infer a list of interactions between the $p$ genes and their predicted regulators, thereby constructing the gene regulatory network. However, this method has limitations, such as its inability to predict self-regulating genes. [1]

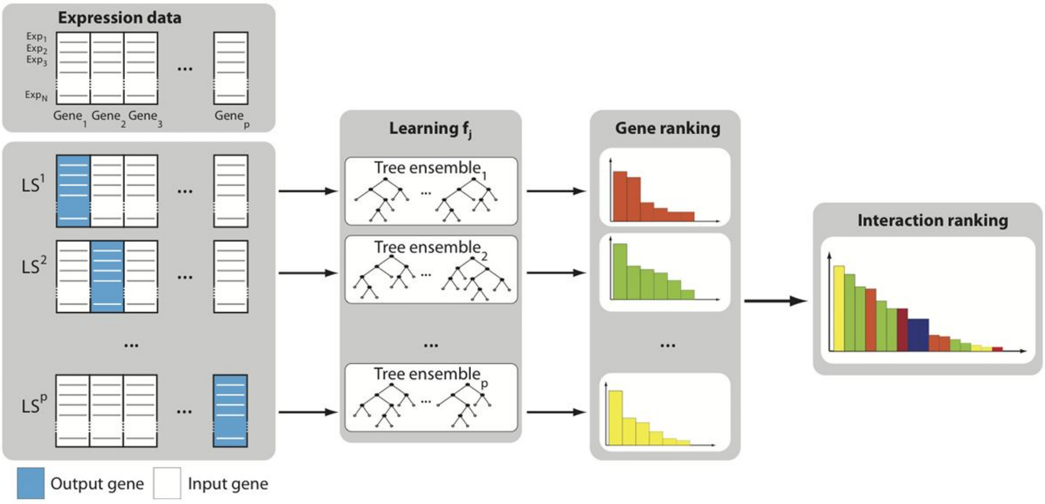


Figure 1.6: *`Genie3` procedure. Expression data is divided into p learning samples, each corresponding to a different gene. Models are trained on these samples and rankings are determined using global feature importance methods [1].*

The global feature importance used by `Genie3` is known as the mean decrease in impurity. A detailed review of this method is provided in section 2.1.1.

### 1.3.3 Cell-specific and cell-type specific network inference

Global gene regulatory network inference deduces genes interactions and networks across the entire dataset. However, in certain applications, it is useful to infer networks specific to particular regions of the dataset.

One such problem of interest is the **inference of gene regulatory networks for individual cells within the dataset**.
Advances in single-cell sequencing technologies have provided a wealth of single-cell data, enabling identification of more complex and non-linear dependencies between cells. Single-cell regulatory network inference aims to leverage this data to capture differences between cells under varying environmental conditions.
Several methods have been proposed in the literature to infer single-cell gene regulatory networks [10, 11, 12, 13].

This thesis proposes an approach based on `Genie3`, but instead of using a global feature importance method, the algorithm uses local feature importance methods. These methods function similarly to global feature importance algorithms but assign importance scores to features of each sample. As in `Genie3`, interactions and networks can then be inferred from these importance scores for each sample (in this case, each cell).

Another application of this approach is to aggregate local feature importance scores based on specific criteria. A common criterion is to **group cells by cell type**. Cell-type-specific gene regulatory network inference thus involves identifying networks specific to particular cell types based on single-cell gene expression data.

The methodology followed in this thesis to solve this task is to average single-cell gene regulatory networks for cells of the same type. The objective is to capture common interactions across cell types as in [11], while using the entire dataset rather than applying global methods to each type individually.

# Chapter 2

# Methods

As outlined in subsections 1.3.2 and 1.3.3, the methodology for identifying regulatory connections among $p$ genes within a network involves applying feature importance methods to $p$ machine learning models. Each model provides information about the key genes that contributes to predict the expression level of the $p^{th}$ gene, i.e. genes that regulates its expression.

Feature importance methods offer a range of approaches to assess the significance of each feature in influencing the behaviour of a machine learning model. These methods can be classified into two categories: global methods, which evaluate the importance of each feature in the behaviour of the model, and local methods, which focus on the importance of each feature in the context of the specific prediction of a sample.

This chapter details the feature importance methods employed, including specific implementations. The two first sections describe global and local feature importance methods, while the final section discusses various normalisation methods used on the feature importance values.

## 2.1 Global Feature importance Methods

Global feature importance methods rank the contribution of each feature to the overall behaviour of a model. This section introduces two such methods. The `Genie3` algorithm, which is commonly used for gene regulatory network inference, uses a particular global feature importance method, called GlobalMDI. Additionally, GlobalMDA, is also detailed.

### 2.1.1 Genie3 feature importance method

`Genie3` uses tree-based models and global feature importance to infer a global gene regulatory network from gene expression data. As described in section 1.3.2, `Genie3` employs a generic feature importance method provided by `scikit-learn`, specifically the mean decrease of impurity (MDI) for regression trees. The method compute the reduction in impurity, $\Delta i(s_t, t)$, at each node where a particular variable is chosen for splitting [14, 15].

For a given tree $T$, the importance for the variable $X_m$ in predicting output $Y$ is given by

$$Imp(X_m, T) = \sum_{t \in T : v(s_t) = X_m} p(t) \Delta i(s_t, t), \tag{2.1}$$

where $t$ represents internal nodes of the tree, $p(t)$ denotes the proportion of training samples reaching node $t$, and $v(s_t)$ is the variable used for splitting at node $t$.

This measure can be generalised to an entire forest, by averaging the importance over all trees

$$Imp(X_m) = \frac{1}{N_T} \sum_T Imp(X_m, T). \tag{2.2}$$

In regression trees, impurity is defined as the reduction in variance. Therefore, equation (2.1) can alternatively be characterised as the reduction in variance of $Y$ due to all splits

$$Imp(X_m, T) = \sum_{t \in T : v(s_t) = X_m} \#S_t Var(S_t) - \#S Var(S_{\text{true}}) - \#S Var(S_{\text{false}}), \tag{2.3}$$

where $S_t$ is the set of training samples reaching node $t$, and $S_{\text{true}}$ (respectively $S_{\text{false}}$) is the subset for which the test done at node $t$ is true (respectively false) [1].

This method efficiently exploits the tree structure, as feature importance is computed during tree growth, making it computationally inexpensive. However, the instability of the regression trees (section 1.1.2) can affect these measures, though this issue is strongly mitigated by the forest structure [16].

### 2.1.2 GlobalMDA

Another global feature importance method is the mean decrease of accuracy (MDA), which was originally developed for random forest models in [14], and later adapted to be model-agnostic, i.e. to work on all machine learning models [17].

The global MDA of a model is computed through the use of permutations of the feature's values, and is based on the following algorithm.

1. Computation of the model loss $L(Y, f(X)) = (Y - f(X))^2$,
2. Construction of a matrix where feature $i$ is permuted,
3. Computation of the model loss on the permuted data $L_p(Y, f(X_p)) = (Y - f(X_p))^2$,
4. Computation of $i^{th}$ feature importance as the difference between $L_p$ and $L$.

This procedure is repeated $n_{\text{permutations}}$ times for each feature. There are two main approaches to consider those permutations: either $f(X_p)$ in the loss function is replaced by the mean of the predictions from the permuted samples, or the mean of the $n_{\text{permutations}}$ differences in the loss function between the permuted and non-permuted samples is used.
The algorithm applies these permutations to a feature $i$ to assess the relationship between the inputs and the output as though $i^{th}$ did not exist. This process effectively "breaks" the dependence between $i^{th}$ feature and the output [16].

However, this method has the drawback of requiring numerous permutations to achieve stable results, which leads to a longer computation time compared to MDI [16].

## 2.2 Local importance Methods

Local feature importance methods rank the contribution of each feature to the behaviour of a specific instance provided to a model.

In this thesis, these methods are employed to determine cell-specific gene regulatory networks by calculating feature importance scores for each individual cell.

### 2.2.1 Shap

`Shap` is a practical implementation of a method for computing Shapley values, as defined in [18].

### Shapley values

Shapley values originate from game theory, where they provide a method to fairly distribute a payout among players. When applied to machine learning, the "payout" corresponds to the model's output, and the "players" are the features. Each feature is assigned a score that reflects its influence on the output.
Given $p$ players $V = \{X_1, ...X_p\}$ and a characteristic function $v : v \to \mathbb{R}$ with $v(\emptyset) = 0$, assessing each possible subset of features, the Shapley value distribution is defined as equation 2.4 [19].

$$\phi_j(v) = \sum_{S \subseteq \{1,\ldots,p\} \setminus j} \frac{|S|!(p - |S| - 1)!}{p!} (v(S \cup \{j\}) - v(S)). \tag{2.4}$$

In this equation, the Shapley value is computed as the sum over all possible coalitions $S$ of features that exclude feature $j$. Each term in the sum is weighted by $\frac{|S|!(p-|S|-1)!}{p!}$, and the expression $(v(S \cup \{j\}) - v(S))$, represent the marginal contribution of feature $j$ to the coalition [20]. The purpose of the characteristic function in evaluating the Shapley value is to map each coalition to a real number.

Thus, Shapley values can be summarised as the weighted average marginal contribution of a feature across all possible combinations of feature [16]. They describe how much each feature contributes to the model's prediction.

Shapley values satisfy several important properties [21, 20, 19].

- **Efficiency**: The total sum of contributions equals the total payout

$$\sum_{X_m} \Phi_v(X_m) = v(V). \tag{2.5}$$

- **Null Player**: A feature that does not contribute to any coalitions has a Shapley value of zero

$$\Phi_v(X_m) = 0 \text{ if } v(S \cup \{X_m\}) = v(S) \text{ for all } S \subseteq V^{-i,j} \tag{2.6}$$

- **Symmetry**: Identical features receive the same contribution

$$\Phi_v(X_m) \geq \Phi_w(X_m) \text{ if } = \Phi_v(X_j) \text{ if } v(S \cup \{X_i\}) = v(S \cup \{X_j\}) \text{ for all } S \subseteq V^{-i,j} \tag{2.7}$$

- **Additivity**: For a game with 2 characteristic functions, the Shapley value of the sum of games is the sum of Shapley values

$$\phi_{j,v_1+v_2} = \phi_{j,v_1} + \phi_{j,v_2} \tag{2.8}$$

Shapley values are unique in that they are the only marginal values that satisfy these properties [20].

To calculate the Shapley value, it is necessary to compute values for all possible coalitions, which is often computationally unfeasible. Various methods such as MonteCarlo sampling, approximations or specific implementations exist to address this challenge [16].

A popular package that introduces several of these methods is `Shap`.

### Shapley Additive Explanations

`Shap` is a method introduced by Lundberg and Lee in 2017 [18]. The `Shap` approach to compute Shapley values treats the model as an additive feature importance method. In this context, the contribution of each feature to the model's output can be expressed as a linear model

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j, \tag{2.9}$$

Let $g$ be the explanation model, and let $z' \in \{0,1\}^M$ represent the coalition vector, where $z'_j = 1$ indicates the presence of feature $j$ in the coalition and $z'_j = 0$ indicates its absence. Here, $M$ denotes the maximum size of the coalition. The feature attribution for a feature $j$ is denoted by $\phi_j \in \mathbb{R}$, which corresponds to the Shapley value of feature $j$ [16].

`Shap` defines its characteristic function for a model $f$ and an instance $x^i$ as shown in equation 2.10 [20].

$$v_{f,x^i}(S) = \int f(x^i_S \cup X_c) d\mathbb{P}_{X_c} - \mathbb{E}(f(X)). \tag{2.10}$$

In this equation, the term $f(x^i_S \cup X_c) d\mathbb{P}_{X_c}$, simulates the features not included in $S$ as random variables. The equation (2.10) thus involves integrating over their distribution, a process known as marginalisation. When equation (2.10) is substituted in the Shapley value equation ((2.4)), the expectation term $\mathbb{E}$ simplifies. The remaining expression calculates a weighted difference in the marginalisation between the set $S \cup \{j\}$ and $S$ for all subset $S$ that do not include feature $j$, capturing the marginal contribution of $j$.

Given the number of possible coalitions increases exponentially with the number of features $p$ ($O(2^p)$), approximating Shapley values become more efficient. In this thesis, a specific variant of `Shap`, known as `TreeShap`, is used.

`TreeShap` exploits the tree structure to accelerate computations. Although there are a high number of potential coalitions, the tree structure restricts the possible outcomes, with many coalitions yielding to the same result at a particular node [20].

It is straightforward to decompose a model's prediction with $p$ features using equation (2.9) [19], as follows

$$\hat{f}(X) = E\{\hat{f}(X)\} + \phi_v(X_1) + \cdots + \phi_v(X_p). \tag{2.11}$$

As an illustration, consider an explanation of a model's output provided in the `Shap` package [22] (see figure 2.1). This example demonstrates how the contribution (Shapley value) of each feature is added to the mean of prediction to determine its specific contribution for a given prediction.

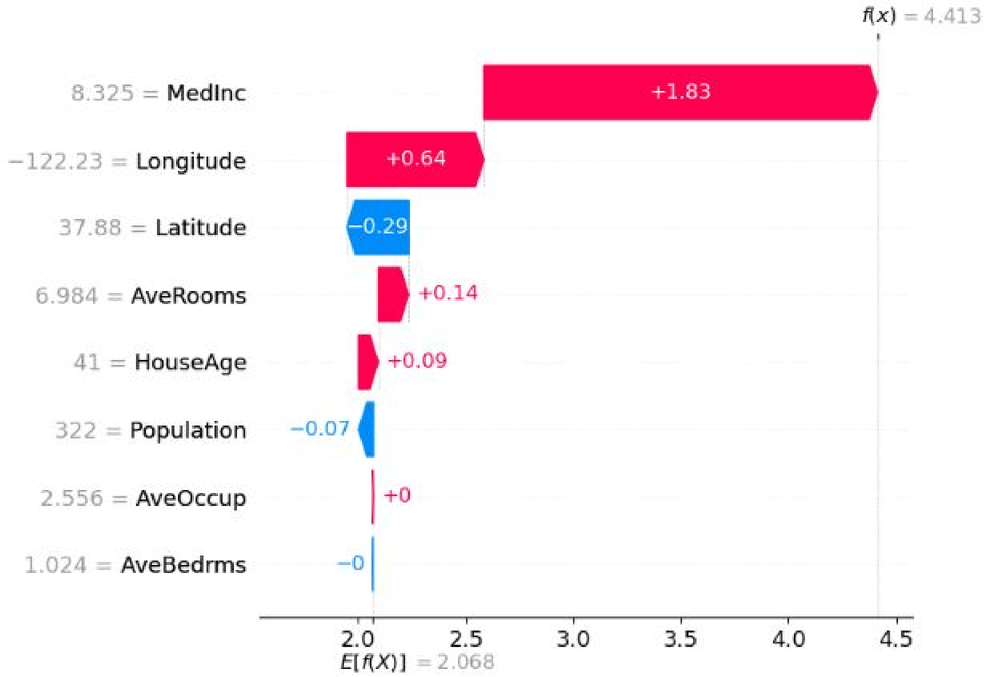


Figure 2.1: *`Shap` decomposition of features contribution to the output. Each feature's contribution adds up to explain the difference between the expected value of predictions and the sample prediction [22].*

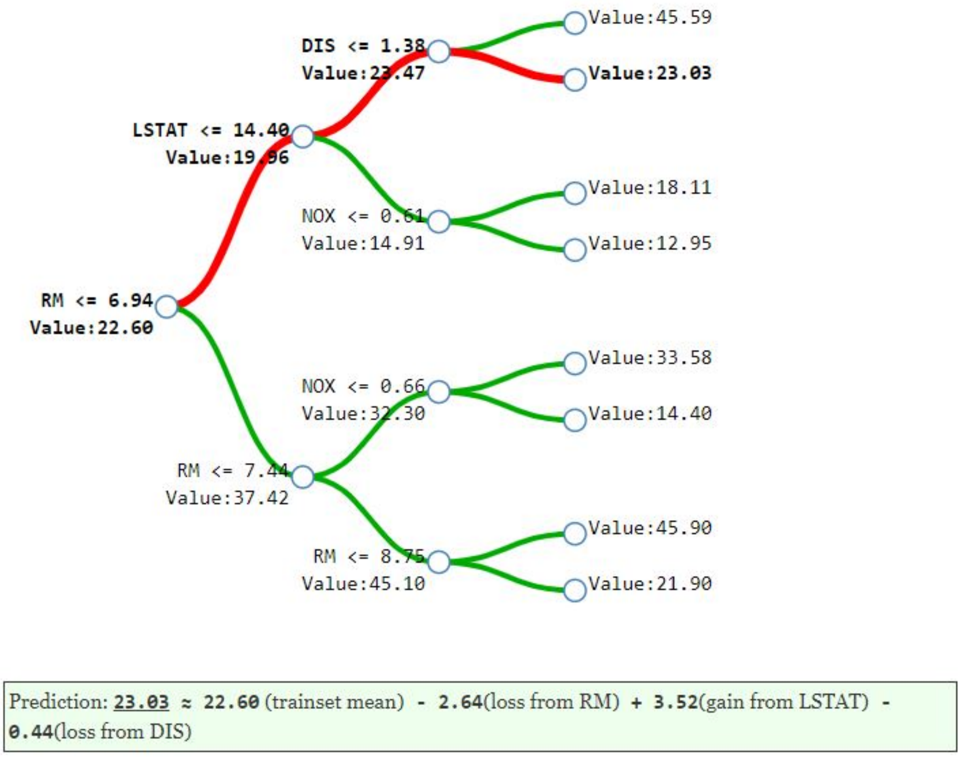
### 2.2.2 Saabas

Saabas is a method detailed in the `TreeInterpreter` package. Like `Shap`, its purpose is to decompose a model's prediction into a sum of a bias term and feature contributions. Given a vector of $N$ features $X$ and $\theta_i$ representing the contribution of the $i^{th}$ feature to the prediction, the prediction $\hat{f}(X)$ is expressed as

$$\hat{f}(X) = bias + \sum_{i=1}^{N} \theta_i. \tag{2.12}$$

Here, the bias term represents the average prediction value over the model's training set samples.

In Saabas, each regression node of the tree, not just the leaves, has an associated "output" value. The contribution of a feature $i$ at a split $k$ is calculated as the difference in value between the node before the split $k$ and the node reached after the split $k$ on feature $i$. The total contribution of the $i^{th}$ feature is the sum of its contribution across all splits where it is used (see figure 2.2) [23].



Prediction: 23.03 ≈ 22.60 (trainset mean) - 2.64(loss from RM) + 3.52(gain from LSTAT) - 0.44(loss from DIS)

Figure 2.2: *Saabas decomposition of feature contributions to the output. Each feature's contribution is the difference in value between the node before the split and the node reached after the split, for all nodes where the feature is used to split* **??**.

Saabas can be viewed as an approximation of `Shap`. Unlike `Shap`, which considers all possible feature coalitions, Saabas uses only one specific ordering of features, defined by the path taken by the particular sample of interest [24].
In the context of random forests, the feature importance scores are averaged across all the trees in the forest.

### 2.2.3 LocalMDI

LocalMDI is a method derived from the global MDI method, as presented in [15] and implemented in [25]. It can be seen as a decomposition of the global measure.

Let us indicate a specific instance of the input variables by $\mathbf{x} = (x_{,} \cdots, x_p)^T$, where $x_j$ represents the value of variable $X_j$. For a given instance $\mathbf{x}$, the local MDI importance $Imp(X_m, \mathbf{x})$ of a variable $X_m$ with regard to $Y$ is defined as

$$Imp(X_m, \mathbf{x}) = \frac{1}{N_T} \sum_{t \in T : v(s_t) = X_m \wedge \mathbf{x} \in t} i(t) - i(t_{X_m}), \tag{2.13}$$

where $X_m$ is used to split, $t_{X_m}$ is the successor of node $t$ followed by $\mathbf{x}$ in the tree, and $i()$ is the impurity function used. The outer sum is over the $N_T$ trees in the ensemble, the inner sum is over all nodes that are traversed by $\mathbf{x}$.[15].

This measure computes the significance of the specific path taken by the sample of interest, focusing on how the variables used in splits contribute to the prediction for that particular sample. The term $i(t) - i(t_{X_m})$ quantifies the differences in impurity between nodes on the path taken by the sample through the trees. A variable is considered highly important if it consistently leads to a significant reduction in impurity across the majority of trees.

The global MDI measure, from which LocalMDI is derived, can be obtained by summing the values as

$$Imp(X_m) = \frac{1}{N} \sum_{i=1}^{N} Imp(X_m, \mathbf{x}^i), \tag{2.14}$$

where $\{(\mathbf{x}^1, y1), \cdots, (\mathbf{x}^N, y^N)\}$ represent the $N$ learning samples [15].

Similar to its global version, LocalMDI is highly efficient because the impurity measures are already computed and used during the tree-growing process.

### 2.2.4 LocalMDA

The localMDA method is a decomposition of globalMDA defined in [14]. Given a loss function $L()$, the labels $Y$, the prediction $f(X)$ of the samples $X$ by a model $f$, and a sample $X_p$ where the feature $p$ has been permuted in all samples by the value of another sample, the globalMDA feature importance for the $p^{th}$ feature is defined as

$$Imp(p) = L(Y, f(X)) - L_p(Y, f(X_p)). \tag{2.15}$$

For a specific sample and its label $(\mathbf{x}^i, y^i)$, the localMDA feature importance is defined as

$$Imp(p, \mathbf{x}^i) = L(y^i, f(\mathbf{x}^i)) - L_p(y^i, f(\mathbf{x}_p^i)). \tag{2.16}$$

As in the global method, the permutations are repeated multiple times. This thesis explores two possible approximations of the decrease in accuracy. The first approach replaces the prediction $f(\mathbf{x}_p^i)$ in the loss function with the mean of the predictions of the permuted samples, while the second approach computes the mean over the

$n_{\text{permutations}}$ differences in loss between permuted and non-permuted samples will be taken. As in globalMDA, the permutation is introduced to simulate the removal of the feature $p$ from the sample.

Like global MDI, localMDA can be seen as a decomposition of globalMDA:

$$Imp(p) = \frac{1}{N} \sum_{i=1}^{N} Imp(p, \mathbf{x}^i), \tag{2.17}$$

where $\{(\mathbf{x}^1, y1), \cdots, (\mathbf{x}^N, y^N)\}$ represents the learning sample of $N$ examples used to grow the ensemble of trees.

Three different implementations of the localMDA method were developed. The first one applies the model-agnostic method directly to the random forest model, the second one exploits the ensemble structure of a random forest, and the third directly uses the tree structure to compute the loss of accuracy, as described in [26].

### LocalMDA based on perturbations of the forest

The application of the localMDA model-agnostic methods treats consider the random forest regressor as the model. The loss function (in this case, the mean squared error loss) is computed based on the prediction of the forest, when permuting values of the $p^{th}$ feature, for each feature.
To reduce sensitivity to a particular permutation, 100 permutations were conducted for each feature per sample. However, this setting leads to high computational time. In this method, the mean of the prediction from permutations is used to define the prediction $f(\mathbf{x}_p^i)$.

### LocalMDA based on perturbations of the trees

The main drawback of the permutation-based methods is the high computational complexity involved in computing predictions for each permutation.
An alternative is to take advantage of the ensemble structure of the forest. The prediction $\hat{y}$ of a sample $\mathbf{x}$ by a forest model $T$ with respect to its trees $t$ is given by

$$\hat{y}(T, \mathbf{x}) = \frac{1}{N_t} \sum_{t \in T} \hat{y}_t(\mathbf{x}). \tag{2.18}$$

Using the formalism of equations (2.18) and (2.16), another measure of the average decrease in accuracy is defined, considering a different permutation for the sample

$\mathbf{x}_p^i$.

$$Imp(p, \mathbf{x}^i) = \frac{1}{N_t} \sum_{t \in T} L(y^i, f(\mathbf{x}^i), t) - L_p(y^i, f(\mathbf{x}_p^i), t), \tag{2.19}$$

where $L(y^i, f(\mathbf{x}^i), t)$ (respectively, $L_p$) represents the loss function evaluated for each regression tree separately for the non-permuted (respectively, permuted) sample, and $\mathbf{x}_p^i$ is the $i^{th}$ sample where feature $p$ is permuted with another value, -different for each tree.

Here, the permutations differ for each tree, and the mean of the difference in loss values across trees is computed for the permuted and non-permuted samples.
This method requires only one passage through all the trees, rather than of $n_{\text{permutations}}$ passages, which reduces computational time. However, the forest must contain a sufficient number of trees to provide stable results. In this thesis, forests composed of 1000 trees were used, and the results were available for interpretation.

### LocalMDA based on tree structure

Another method, as described in [26], takes advantage of the tree structure itself. The goal of the permutations mentioned earlier is to simulate the removal of a feature. Another approach to achieve this is by modifying the path that samples follow within the trees.

The localMDA importance value of a feature $p$ for a sample $\mathbf{x}^i$ using a regression tree model $t$, can be defined as

$$Imp(p, \mathbf{x}^i, t) = \sum_{v \in V} L(y^i, g_v(\mathbf{x}^i, p, t), p, t), \tag{2.20}$$

where $V$ is the set of leaves in the tree where the samples $\mathbf{x}^i$ could end up if, at each node where the feature $p$ is tested, the sample is propagated on both sides.

To quantify the effect of removing the feature from a tree, the value $g_v(\mathbf{x}^i, p, t)$ is computed for each leaf where the sample could end up:

$$g_v(\mathbf{x}^i, p, t) = \hat{y}_v \prod_{k \in K} \frac{N_{k,v}}{N_k}, \tag{2.21}$$

where $K$ the set of nodes where a split occurs on $p$, and $N_{k,v}/N_k$ corresponds to the number of samples that followed the path leading to $v$ after node $k$ where $p$ was tested, relative to the number of samples reaching node $k$. The value $\hat{y}_v$ corresponds to the output value returned by the leaf $v$.

For a forest $T$, the measure can be summarised as

$$Imp(p, \mathbf{x}^i) = \frac{1}{N_t} \sum_{t \in T} L(y^i, f(\mathbf{x}^i), t) - L_p(y^i, f(\mathbf{x}^i_p), t) \tag{2.22}$$

No implementation of this method was previously provided. The implementation developed in this thesis was based on the information provided in [26] and algorithms used to traverse decision trees, as defined in [27].

### 2.2.5 LES values

The FESP (Fair-efficient-symmetric-perturbation) value and the ES (Equal-surplus) method were developed as fair alternatives to Shapley values for solving the feature importance attribution problem [28]. These methods belong to the family of LES (Linear-Efficient-Symmetric) values. Their primary goal is to avoid the exact computation of Shapley values.

These values satisfy the key properties of Shapley values: additivity, efficiency, and symmetry, as well as monotonicity, covariance, non-negativity. They also preserve additive games, and marginalism as described in [29].

The LES values, denoted as $\Psi^A$, can be expressed as

$$\Psi_i^A(v) = \sum_{k=1}^{n} \left[ \sum_{i \in S, |S|=k} \frac{(n-k)!(k-1)!}{n!} \left[ A(k)v(S) - A(k-1)(S - \{i\}) \right] \right]. \tag{2.23}$$

From equation 2.23, it can be observed that the Shapley value is a specific case of these LES values.
Specificallly, considering $\Gamma$ as the set of $n{+}1$-vectors of real numbers $A = (A(k))_{k=0,\cdots,n}$, such that $A(0)$ is a fixed real number and $A(n) = 1$, and $\Gamma$ as the $2^n - 1$ dimensional linear space of all $n$-person games on $N$, a finite collection of $n$ players [29],

$$\begin{gathered} \forall v \in \Gamma, \forall i \in N, \forall A \in \Omega, \Psi^A(v) = Shap(v^A), \\ \text{where } v^A(S) = A(k)v(S) \text{ for each } S \text{ such that } |S| = k. \end{gathered} \tag{2.24}$$

Thus, Shapley values represent the particular case where $A(1), ..., A(n) = 1$. Another value within this framework is the Equal-Surplus value [30], characterised by

$A(1) = n - 1$ and $A(n) = 1$, and 0 otherwise. In this context, the term in $A(n) = 1$ represents the contribution of the specific feature, while $A(1) = N - 1$ corresponds to the ES value, which uses the difference between the total coalition and the sum of individual feature contributions. This term, however, remains constant across all features [28].

The three methods discussed in this section use the source code provided in the GitHub repository [31]. However, since the model is a Random Forest model and its scoring function is not well-defined for individual samples, this package was modified to use the least square error as evaluation metric for predictions.

**FESP**

To address the issue of this constant term across all features, [28] proposed a model that considers not the equal surplus value, but the value of $v(N \setminus \{j\})$ (extreme coalitions model). This approach respects efficiency, symmetry, and fairness, as demonstrated in [28].

**ES**

The ES method strictly adheres to the evaluation of ES values as described above[28].

## 2.3 Normalisation Methods

This section discusses various normalisation and scaling methods applied to feature importance values obtained after running the algorithms.

### 2.3.1 L1 normalisation

L1 normalisation transforms the values of a vector by dividing each value by the sum of the absolute values of the entire vector, as shown below:

$$\mathbf{x_{l1}} = \frac{\mathbf{x}}{\sum_i |x_i|}.$$

The primary advantage of the L1 normalisation is its ability to preserve the sparsity of the data when the data is already sparse. Additionally, it is less sensitive to outliers compared to others techniques such as L2 normalisation.

### 2.3.2 L2 normalisation

L2 normalisation scales the values of a vector such that the sum of the squared values equals one. Specifically, each value is divided by the square root of the sum of the squared values of the vector:

$$\mathbf{x_{l2}} = \frac{\mathbf{x}}{\sum_i x_i^2}.$$

While L2 normalisation is more sensitive to outliers due to the squaring of values, it tends to smooth the data. This may lead to a reduction of the sparsity of the vector.

### 2.3.3 Max normalisation

Max normalisation divides the values of a vector by its maximum value:

$$\mathbf{x_{max}} = \frac{\mathbf{x}}{x_{max}}.$$

This ensures that the maximum value of the vector is always one. However, this method is highly sensitive to upper outliers. Indeed, those can be incorrectly identified as the maximum value, thereby distorting the scaling.

### 2.3.4 Min-max scaling

The min-max scaling technique scales the values of a vector over an interval between 0 and 1, where 0 corresponds to the minimum value and 1 to the maximum value:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}.$$

Despite its utility,this scaling is highly sensitive to outliers, as any outlier can be mistaken for the minimum or the maximum value.

# Chapter 3

# Datasets

In this chapter, different datasets will be reviewed, together with their role in the master thesis. Four types of datasets will be investigated. A first dataset, Friedman1, will be used as a generic dataset to evaluate the methods on a simple problem. Two other types of datasets, Dyngen-generated datasets and cell-type datasets, are synthetic datasets used to evaluate the methods on the specific single-cell gene regulatory networks and cell-type gene regulatory networks inference problem. Finally, a real dataset without ground truth networks will be investigated.

For synthetic datasets, the mechanisms behind their generation will be described, as well as the methodology used to generate them. As the real dataset is concerned, details about its origin will be given.

## 3.1 Friedman1

Friedman dataset finds its origins in the Friedman regression problem [32]. This dataset contains $n$ independent features with a uniform distribution between 0 and 1. The particularity of the dataset is that only the 5 first features are taken into account to produce the output $y$. All the remaining $n - 5$ are considered as independent of $y$. The output $y$ is generated using the relationship 3.1 [33]

$$y(\mathbf{x}) = 10 \sin (\pi \times x_1 \times x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5. \tag{3.1}$$

In this thesis, the dataset will be generated using the `make_friedman1` function of the `scikit-learn` library. In particular, 1000 samples will be drawn composed of 105 features each without noise addition.

The relationship has the advantage to be non-linear and easily defines the features related to the output. This dataset will be used in this thesis to make a first comparison of our methods and to verify their working.

## 3.2 Dyngen

This section will focus on Dyngen, a multi-modal simulation engine for single-cell resolution research on dynamic cellular processes [34]. Dyngen will be used to generate gene expression levels datasets with available ground truth regulatory networks for each cell of the dataset.

### 3.2.1 Generation of Gene expression levels datasets using Dyngen

Dyngen is a simulation tool used in the context of single-cell resolution analysis. Single-cell biology is expanding, however, the lack of ground-truth makes quantitative evaluations of methods often impossible. The goal of Dyngen is thus to generate synthetic data with available ground truth [34].

The advantage of Dyngen compared to others single-cell simulators is to focus on the fundamental biology governing these interactions [34].

The inner working of Dyngen is decomposable into 3 main steps [34]:

1. The first step in simulating biological processes is to convert a global gene regulatory network into a series of reactions (translation, splicing, transcription, and regulation).
2. The use of a version of Gillespie's stochastic simulation algorithm (SSA) allows to then simulate individual cells.
3. Finally, single-cell methods are simulated using actual reference datasets.

The use of Dyngen allows validating a lot of computational methods (figure 3.1):

1. Cell-specific network inference, which in this case is the main feature of interest.

2. Trajectory alignment, methods to align the cell's developmental trajectories.

3. RNA velocity methods, methods that estimate the RNA velocity.

4. Trajectory inference, methods that infer the cell developmental trajectories.

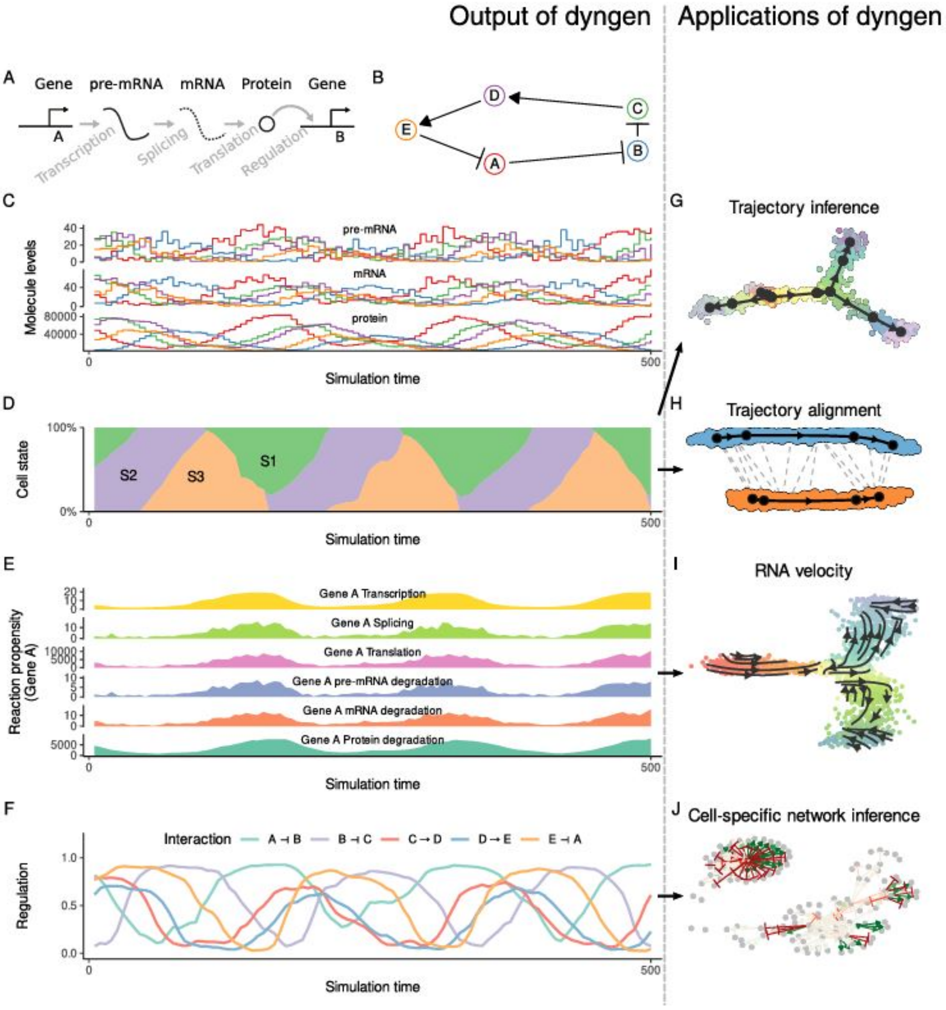


Figure 3.1: *Review of Dyngen functionalities. Dyngen generates single-cell datasets used to validate trajectory alignment and inference methods, cell-specific network inference methods, and RNA velocity estimation methods [34].*

### 3.2.2 Derivation of ground-truth single-cell GRN

In this thesis, the Dyngen functionality that will be used is its ability to label, for each cell and its gene expression levels, a ground truth single-cell regulatory network.

Dyngen allows defining all cell-specific regulatory interactions, computing the effect of a regulator $R$ on a target $T$, using the role that $R$ plays in the transcription of $T$'s propensity function in relation to other regulators [34]. In particular, it can be defined as

$$regeffect_G = \frac{proptrans_G(S) - proptrans_G(S[R \leftarrow 0])}{xpr_G}. \tag{3.2}$$

Where $proptans_G$ is the propensity of transcription of a gene $G$. This last depends on transcription factors and thermodynamic models. The numerator of the expression is thus the difference in propensity when $R$ is set to 0, weighted by a term $xpr$ depending on pre-mRNA production rate of $T$. Pre-mRNA is the messenger RNA before final stages of transcription [34].

By computing all regulatory effects, cell-specific networks can be deduced. The regulatory effects have values between -1 and 1, representing total inhibition or activation respectively, where 0 represents inactive regulation.

### 3.2.3 Methodology & datasets obtained

Dyngen simulator offers a wide range of various base networks on which simulations can be run and datasets made. In [34], 42 distinct datasets were tested on single-cell gene regulatory network inference, with 14 different backbones.

The backbones are the following:

1. Bifurcating,
2. Bifurcating Converging,
3. Bifurcating Cycle,
4. Bifurcating Loop,
5. Binary Tree,
6. Branching,
7. Consecutive Bifurcating,
8. Trifurcating,

9. Converging,
10. Cycle,
11. Disconnected,
12. Linear,
13. Linear Simple,
14. Cycle Simple.

In this thesis, one dataset of each type of backbones available on Dyngen will be investigated. A backbone is a base network, from which a dataset can be generated with slight variations of the network.
Each dataset is generated with the backbone as network. To the backbone is added 10 housekeeping genes and 15 target genes. Housekeeping genes are genes with a high expression value but not regulated by the main network. Target genes are genes regulated by the regulatory network defined and by other genes. Each dataset is made of 1000 cells.

The analysis will focus on the presence or absence of regulatory links, rather than their strength or nature. Therefore, the ground truth will be represented in the form of a matrix of all possible regulations. This matrix will contain binary values: 0 indicating the absence of regulation and 1 indicating the presence of regulation. The directions of these links will also be considered. Indeed, if a network specifies a regulator $g$ and a target $t$, the only regulation defined as present will be $g \to t$ and not $t \to g$.

## 3.3 Cell-type datasets

As described in section 1.3.3, another goal of this thesis was to take advantage of models trained on many cell types to better describe common interactions, while still describing particular interactions of each cell-type network. This specific work can be done by aggregating local feature importance values of cells of the same type. To evaluate this work, a dataset of expression levels data generated from different ground truth networks, but with common interactions, was needed. This one was constructed by permutations of a baseline network, from which gene expression values were deduced using the BoolODE generator.

### 3.3.1 Mimic cell-type representation through permutations

The goal is to compare various cell-types sharing common and different interactions. A simple way to generate similar networks was to permute the edges of the reference network randomly with a probability $p$, as explained in [11] (figure 3.2). These edges will eventually be connected to other genes already present in the network, or to genes not previously present in the network.

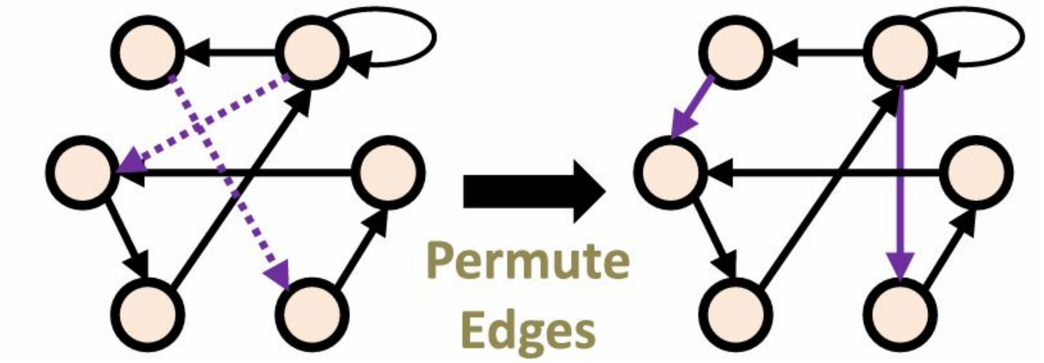


Figure 3.2: *Example of edge permutation of a reference network to generate a similar network [11].*

### 3.3.2 BoolODE

BoolODE is a single-cell gene expression levels generator developed by [9]. It takes as input a boolean model, and uses a system of stochastic differential equations to generate gene expression values. The use of this generator was preferred here to Dyngen, because it was specially designed to compare GRN inference algorithms and preferred to Dyngen in [9]. BoolODE framework is based on GeneNetWeaver framework [35], but with differences to better match existing biological processes (to create variation of the expression profiles, capture single-cell expression trajectories and create less dense regulatory subnetwork).

### 3.3.3 Methodology & datasets obtained

Three different datasets were built, each one with 10 distinct types derived from a different baseline network.

The reference networks were not built from scratch, but based on existing reference networks, especially reference networks used by simulators. In this case, the three networks chosen were networks from Dyngen already generated in the Dyngen datasets.

These three networks were chosen because they contained fewer genes. Indeed, considering permutations, the generated network can sometimes become more structurally complex, and can lead to unpracticable computations for the simulators.

These networks were the following:

1. Bifurcating cycle.
2. Converging.
3. Linear.

Then, for each one, each edge was permuted with a probability $p = 0.2$, to generate a new subnetwork, and the operation was repeated 10 times. The networks were then used as reference for the BoolODE simulator, generating gene expression data. The corresponding expression values matrix were labelled with their type $(1, ..., 10)$ and aggregated to form a unique dataset.

## 3.4 CEDAR dataset

CEDAR dataset is a real dataset composed of a normalised gene expression levels matrix for 27 types of cells of peripheral blood, from different individuals. No ground truth is available concerning the cell-type or global GRNs, however, a list of particular genes of interest and a list of particular genetic markers for each individual are available.

### 3.4.1 Methodology and datasets obtained

Since the dataset has no ground truth, it is not possible to use it to compare predicted and true gene regulatory networks. The analysis of the dataset will then consider a comparison of the global and local rankings of the interactions. Since the real dataset contains 16000 genes, the methods will only be applied to a list of genes of interest.

A first analysis will consider a dataset composed of all the genes of interest expression values and the cell-types. For each type, the mean of the cell feature importance rankings will be computed on the dataset with the 27 types mixed. It will be compared to the global feature importance rankings computed on each of the 27 single type datasets. The dataset is composed of the data of 200 patients. For each of them, a cell of each type exists, with 174 gene expression levels (which are only the genes of interest present in all cell-types) and a categorical value for each sample corresponding to its cell type.

A second dataset will be generated, with all the genes of interest expression levels and a list of genetical markers present for each individual. The goal will be to consider the correlation between the rankings of different samples to observe if the markers affecting a patient have an influence on the generated interactions.
The dataset is composed of the same patients, cells and gene expression values. However, instead of a categorical value indicating the type, it contains categorical values for each sample corresponding to three values of presence of each genetic marker (0 meaning the absence, 1 or 2 the presence).

# Chapter 4

# Results & discussion

In this section, the outcomes of local techniques on each dataset, together with the associated methodology for obtaining these outcomes, are presented. Using the Friedman1 dataset, a preliminary analysis looks at the mean area under the receiver operating characteristic curve (AUROC) and the area under the precision recall curve (AUPR) and compare them to global rankings. Using this dataset, Spearman's correlation is also investigated in order to gauge method similarity. Next, using synthetic datasets for single-cell and cell-type GRN inference, the AUROC and AUPR measures are evaluated.

Normalisation's effects are also discussed. The actual dataset is examined in the final section, where the feature rankings of one local approach (localMDI) and one global method (`Genie3`) are contrasted. By calculating the Spearman's correlation between the local feature important scores of patients with and without genetic markers, an additional analysis is performed to compare the effect of genetic markers on the feature importance ratings.

## 4.1 Friedman 1 dataset

Friedman 1 dataset, described in section 3.1, will be a useful synthetic dataset to compare our methods in the context of a simple regression problem. Since the exact features used to generate the output are known, the dataset is exactly what is needed to evaluate the efficiency of the methods.

### 4.1.1 Methodology

Our goal will be to compute feature importance of each feature to predict the output $y$. The mean AUPR and AUROC values will allow comparing the average of local importance scores to global scores, to see if local methods perform on a simple regression problem. The impact of normalisation will also be studied in this contest. Another interest will be brought to the measure of the correlation between the feature importance values to determine similarities between local techniques.
In particular, the evaluations will be made using a RandomForestRegressor model, with $n_{estimators}$ equals to 1000. Others parameters will be kept as default.

### 4.1.2 AUROC & AUPR

In this section, all AUROC and AUPR curves were computed with corresponding functions from `scikit-learn` library. AUPR and AUROC metrics are described in the section 1.2.1.
In the case of Friedman dataset, all outputs are determined by the five first features. In order to compute the AUROC & AUPR scores of methods, will be constructed a vector of the size of the $n$ features. It contains 0, the boolean False value, everywhere excepted for the five feature of importance, which will be equal to 1, the boolean True value.

In the case of local feature importance methods, for each sample the AUROC & AUPR scores will be calculated comparing the sample's feature importance vector and the vector described above. Then, the mean of the scores over all the 1000 samples will be taken. For the global methods, the scores will simply be compared to the vector.
Following the above methodology:

| Local methods | meanAUROC score |
|---|---|
| Local MDI | 0.996 |
| Shap | 0.992 |
| Saabas | 0.528 |
| LocalMDA - forest perturbations | 0.979 |
| LocalMDA - tree perturbations | 0.978 |
| LocalMDA - tree structure | 0.656 |
| FESP | 0.243 |
| ES | 0.448 |

Table 4.1: *Mean AUROC scores for local feature importance methods for the detection of the 5 important features in the generation of the output y in Friedman 1 dataset.*

| Local methods | meanAUPR score |
|---|---|
| Local MDI | 0.926 |
| Shap | 0.88 |
| Saabas | 0.496 |
| LocalMDA - forest perturbations | 0.831 |
| LocalMDA - tree perturbations | 0.866 |
| LocalMDA - tree structure | 0.572 |
| FESP | 0.186 |
| ES | 0.398 |

Table 4.2: *Mean AUPR scores for local feature importance methods for the detection of the 5 important features in the generation of the output y in friedman 1 dataset.*

| Global methods | AUROC score |
|---|---|
| Global MDA | 0.99 |
| Genie3 MDI | 0.99 |

Table 4.3: *AUROC score for global feature importance methods for the detection of the 5 important features in the generation of the output y in friedman 1 dataset.*

| Global methods | AUPR score |
|---|---|
| Global MDA | 0.94 |
| Genie3 MDI | 0.94 |

Table 4.4: *AUPR score for global feature importance methods for the detection of the 5 important features in the generation of the output y in friedman 1 dataset.*

Overall, the **2 global methods performs better than the local methods**. That can be easily explained by the design of the problem and the dataset. Indeed, since all samples depend on the same 5 features, a global method, whose goal is to find the overall most important features, will necessarily be more suited to this problem. The goal of the analysis here was to find if some methods, applied locally, by taking their mean AUPR/AUROC scores, could challenge these methods on this type of problem.

**Three local methods provides superb results** (Local MDI, `Shap`, LocalMDA based on forest and tree perturbations) on the friedman1 dataset. They all shares a AUROC score between 0.97 and 0.99, which relates to a nearly perfect classification of important features (table 4.1), as the global methods (table 4.3). Their AUPR score between 0.83 and 0.93 provides information that these methods were excellent at finding the few features influencing the output, but they also did not take too many other features into account, even considering a single sample (table 4.2). The global methods, with a score of 0.94, are even more precise (table 4.4).

Methods as Saabas, ES and LocalMDA based on the tree structure results are a bit less interesting, but have to be tested on other types of data, with more differences between the samples to highlight their performance on a more heterogeneous dataset. With a AUROC score between 0.44 and 0.65, their performance is close to the one of a random classifier (table 4.1). Their AUPR score between 0.39 and 0.6 highlights their lack of precision, even on a simple problem (table 4.2).

FESP method seems to perform poorly (tables 4.2 and 4.1). Both AUROC and AUPR scores are around 0.2, which means that they are not able to distinguish the features of interest.

### 4.1.3 Spearman correlation

Spearman correlation metric is described in the section 1.2.2. Another aspect that can be studied, apart from how the rankings of the features correspond to the ground truth computing the AUPR and AUROC, is how the local rankings correlate between themselves.

Spearman correlation was hence computed between each pair of matrices of feature importance values. It was computed comparing each sample feature importance values, and the mean of the correlations was taken. The method used implies the use of the `scipy` function `spearmanr`.

| Correlation | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | / | 0.494 | 0.011 | 0.57 | 0.48 | 0.11 | 0.16 | 0.17 |
| 2 | 0.494 | / | 0.009 | 0.67 | 0.369 | 0.179 | 0.18 | 0.05 |
| 3 | 0.011 | 0.009 | / | 0.007 | 0.008 | 0.001 | 0.05 | 0.01 |
| 4 | 0.57 | 0.67 | 0.007 | / | 0.45 | 0.235 | 0.25 | 0.06 |
| 5 | 0.48 | 0.369 | 0.008 | 0.45 | / | 0.616 | 0.18 | 0.05 = |
| 6 | 0.11 | 0.179 | 0.001 | 0.235 | 0.616 | / | 0.18 | 0.05 |
| 7 | 0.16 | 0.18 | 0.05 | 0.25 | 0.18 | 0.18 | / | 0.37 |
| 8 | 0.17 | 0.05 | 0.01 | 0.06 | 0.05 | 0.05 | 0.37 | / |

Table 4.5: *Mean Correlation values between matrix of feature importance values generated by local methods on friedman 1 dataset. The methods identify as the following: 1. LocalMDI, 2.* `Shap`*, 3. Saabas, 4. LocalMDA - forest perturbations, 5. LocalMDA - tree perturbations, 6. LocalMDA - tree structure, 7. FESP, 8. ES*

From table 4.5 can be made the following observations.
In general, **many high correlation scores concern methods that both well-performed** on the dataset. Considering the scores of the 4 methods that were quoted above, methods (1,2,4,5), have the highest correlation scores of the table between themselves, ranging from 0.36 to 0.67.
However, a few observations of correlations not related to the performance of the methods can be made.

It is clear that **LocalMDA different implementations (4,5 and 6) are rather well correlated**. Indeed, even if the method based on tree structure has very low correlation scores (0.001–0.18) with most of the local methods, it shares rather high scores with the 2 other localMDA implementations (0.23 and 0.616). It has the second higher score of the table with the localMDA implementation with perturbations on the trees, while having lower scores of AUPR/AUROC. It thus validates that the implementation does affect the results, but that they are still correlated.

**Saabas, ES and FESP** local methods seem to give very singular results.

For the first, by examination of the different correlations sample-wise, the low correlation mean is due to the average of very distinct scores. Indeed, Saabas has for some samples a very high correlation with well-performing methods, but for some of them were inversely-correlated. Taking the means of highly positive and negative values gave us this particular score nearly null.

The 2 others methods seem to be different from the others but are correlated. Indeed, ES seems to share similarities with FESP that it does not share with other methods. While its scores are consistently below 0.1, the correlation with FESP is 0.37.

### 4.1.4 Impact of FI values normalisation

In the Friedman 1 dataset, the normalisation methods applied on each sample did not change any of the results. However, it can be seen on the next datasets that these had an influence.

A possible cause is that the metrics evaluated here are produced by taking an average of all the samples, and that samples well-classified were clear samples with very distinct scores, while misclassified samples were not impacted by a change in their values. On more complex problems such as single-cell gene network inference, normalisation could occur more changes in classification.

### 4.1.5 Discussion

Friedman1 dataset was a global feature importance problem where we could test the performance of local methods on each sample and see their mean performance, compared to global methods. A comparison of AUPR and AUROC scores was the approach chosen to assess the methods (figure 4.1)

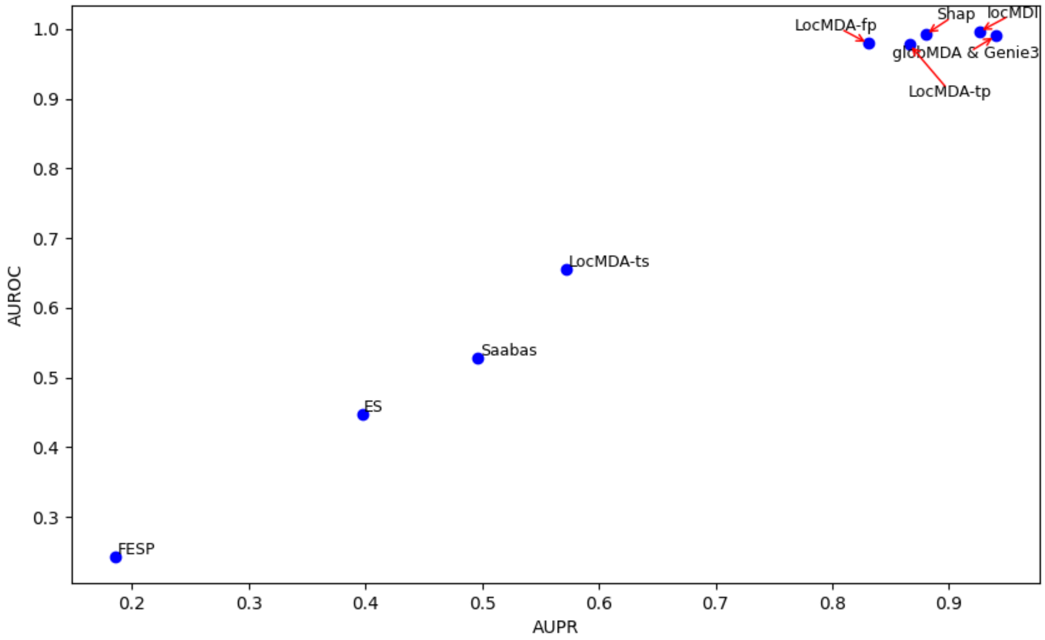


Figure 4.1: *mean of meanAUROC/meanAUPR plot for local methods and global methods on Friedman 1 dataset*

On this dataset, it could be shown that four methods (LocalMDI, `Shap`, LocalMDA based on perturbations) showed significant results. Performance of the others methods remained low, but since the problem was designed for global methods, it is still interesting to study their performance on datasets with more differientated samples. FESP method was however reports catastrophic scores.

The mean of local methods values is thus comparable to the use of global methods in finding a global set of interactions on a simple problem. This result enables the test of local techniques on more complex problems such as cell-specific and cell-type GRN inference.

## 4.2 Dyngen datasets

The objective of this section is to benchmark local feature importance methods on a problem concerning GRN inference, especially cell-specific GRN inference.

The Dyngen simulator allows generating a single cell gene expression datasets from a reference regulatory network, and extracts for each cell a cell-specific regulatory network. Dyngen generated datasets can then be used to evaluate the results of local variable importance methods to predict single-cell GRNs.

### 4.2.1 Methodology

The local inference methods will be applied as described in section 1.3.3. For each dataset of $n$ samples and $g$ genes generated by Dyngen in section 3.2, $g$ random forest regressors will be trained. Each of them is trained on a dataset where the $k^{th}$ model with $k = (1, ..., g)$ is trained on a sub-dataset composed of all genes expression levels excepted the $k^{th}$, whose expression level will be considered as the label.

For each of the models and corresponding sub-datasets, all local feature importance methods and global feature importance described in chapter 2 will be run. These feature scores will then be used to construct a matrix $M'$. If a gene is considered as important to predict the $k^{th}$ gene, then it is probably a regulator of $k$.

The matrix $M$ is a matrix of size $n \times g \times (g - 1)$. It contains boolean true values 1 for each pair $\{i, j\}$ of row $i$ and column $j$ that correspond to a regulation link from $i^{th}$ gene to $j^{th}$ gene, 0 otherwise, for each cell $n$. However, the algorithm is not designed to discover self-regulatory links. Those were deleted from the ground truth and the corresponding elements were erased from the matrix $M$.
The matrix $M'$ is of size $n_{methods} \times n \times g \times (g - 1)$. It contains all the feature importance value computed by the $n_{methods}$, for the $n$ cells, with $k^{th}$, $k = (0, ..., g)$ gene as label and all the remaining as features. This matrix thus measures a score for each value and, for each method, can be compared to matrix $M$ to deduce AUPR and AUROC.

### 4.2.2 AUROC & AUPR

AUPR and AUROC metrics are described in the section 1.2.1. In this section, all AUROC and AUPR curves were computed with corresponding functions from `scikit-learn` library.

The AUPR and AUROC metrics are deduced from the comparison of the ground truth matrix of interactions $M$ and matrix of feature importance $M'$ for each method. For local methods, each sample feature importances are compared to its ground-truth interactions, defined by the Dyngen simulator in the section 3.2. For global methods, the global feature importance are compared to each cell ground truth interactions.

Then, the means of the AUPR and AUROC scores of each of the samples are taken. Since some methods require a lot of computation time, the choice to compute the averages of the metrics on 100 random samples was made. The following scores are obtained for each type of dataset described in section 3.2. Some scores were not reported, because steps of execution of some algorithms could not work with some datasets.

The following tables report the mean across the 14 datasets of the meanAUROC and meanAUPR scores. The choice was made to not report the separated dataset values to improve readiness and interpretability. The detailed values for each dataset are available in appendix A.

| Local methods | meanAUROC score |
|---|---|
| Local MDI | 0.69 |
| `Shap` | 0.7 |
| Saabas | 0.65 |
| LocalMDA - forest perturbations | 0.69 |
| LocalMDA - tree perturbations | 0.64 |
| LocalMDA - tree structure | 0.62 |
| FESP | 0.34 |
| ES | 0.33 |

Table 4.6: *Mean over 14 datasets of meanAUROC scores for local feature importance methods in Dyngen datasets.*

| Global methods | AUROC score |
|---|---|
| Global MDA | 0.8 |
| Genie3 MDI | 0.76 |

Table 4.7: *Mean over 14 datasets of AUROC score for global feature importance methods in Dyngen datasets.*

| Local methods | meanAUPR score |
|---|---|
| Local MDI | 0.048 |
| `Shap` | 0.06 |
| Saabas | 0.071 |
| LocalMDA - forest perturbations | 0.051 |
| LocalMDA - tree perturbations | 0.031 |
| LocalMDA - tree structure | 0.025 |
| FESP | 0.019 |
| ES | 0.024 |

Table 4.8: *Mean over 14 datasets of meanAUPR scores for local feature importance methods in Dyngen datasets.*

| Global methods | AUPR score |
|---|---|
| Global MDA | 0.062 |
| Genie3 MDI | 0.066 |

Table 4.9: *Mean over 14 datasets of AUPR score for global feature importance methods in Dyngen datasets.*

From those results can be observed that **global methods outperforms local methods**. The both global methods provided nearly the same results on the two metrics, but Genie3 seems overall better.

Considering AUROC score (tables 4.6 and 4.7), all the local methods are close, with between 0.18 and 0.06 less score compared to global ones excepted two of them. The worse techniques (FESP, ES) have differences up to 0.47.
**`Shap`, Saabas, LocalMDA (forest perturbations) and LocalMDI** seems to provide the best results. The variants of LocalMDA are close in score but are never the best options. The FESP, ES do not seem to capture the feature importance in this dataset.

Considering AUPR scores (tables 4.8 and 4.9), It can be observed that **the difference between global and local methods is tighter**. A local method, Saabas, even outperformed the global methods by 0.005. Other methods gave good scores (`Shap`, Saabas, LocalMDA (forest perturbations) and LocalMDI) between 0.048 and 0.06. Variants of LocalMDA gave low results (0.025-0.031). FESP and ES did not seem to capture the feature importance.

### 4.2.3 Impact of FI values normalisation

A possible area for improvement was to consider feature importance values normalisation. Especially, the normalisation and regularization techniques described in section 2.3 (l1, l2, max and minmax) were implemented.

The normalisation is applied, for each of the vector of size $g \times (g-1)$ of the matrix $M'$. It means it will result in a normalisation of the interactions in each cell of the dataset. Then, to study the global effect of the techniques, the average of the meanAUROC and meanAUPR for the best normalisation compared to the mean of the base scores of the local methods is reported. The normalisation scores for each method are available in appendix B.

| Local methods \normalisation | normalised | base |
|---|---|---|
| LocalMDI | 0.75 (max) | 0.69 |
| `Shap` | 0.72 (all normalisations) | 0.7 |
| Saabas | 0.65 (l1, l2, max) | 0.65 |
| LocalMDA - forest perturbations | 0.75 (all normalisation) | 0.69 |
| LocalMDA - tree perturbations | 0.70 (l1, l2) | 0.64 |
| LocalMDA - tree structure | 0.68 (l1, l2) | 0.62 |
| FESP | 0.35 (l2) | 0.34 |
| ES | 0.36 (minmax) | 0.33 |

Table 4.10: *mean of meanAUROC score for normalised local feature importance methods for the Dyngen datasets*

| Local methods \normalisation | normalised | base |
|---|---|---|
| LocalMDI | 0.112 (max) | 0.048 |
| `Shap` | 0.098 (max) | 0.060 |
| Saabas | 0.07 (l2) | 0.071 |
| LocalMDA - forest perturbations | 0.089 (max) | 0.051 |
| LocalMDA - tree perturbations | 0.067 (max) | 0.031 |
| LocalMDA - tree structure | 0.067 (max) | 0.025 |
| FESP | 0.020 (l2) | 0.019 |
| ES | 0.024 (max, minmax) | 0.024 |

Table 4.11: *mean of meanAUPR score for normalised local feature importance methods for the Dyngen datasets*

**The normalisation of cells feature importance metric allows an increase in both metrics.** Comparing the AUROC metrics, all normalisations allow an increase of between 0.02 and 0.07 for all the methods, excepted Saabas and FESP. As far as the AUPR metrics are concerned, the increase is the most significant for the normalisation concerning a division of all importance values by the maximum importance value. For some methods, the increase went up to 0.064 (localMDI).

Considering the normalised local importance values, **local methods now outperforms the global techniques**. With the maximum normalisation, `Shap`, localMDI and localMDA based on forest perturbations outperforms global methods in AUPR while still having close results for the AUROC.

### 4.2.4 Discussion

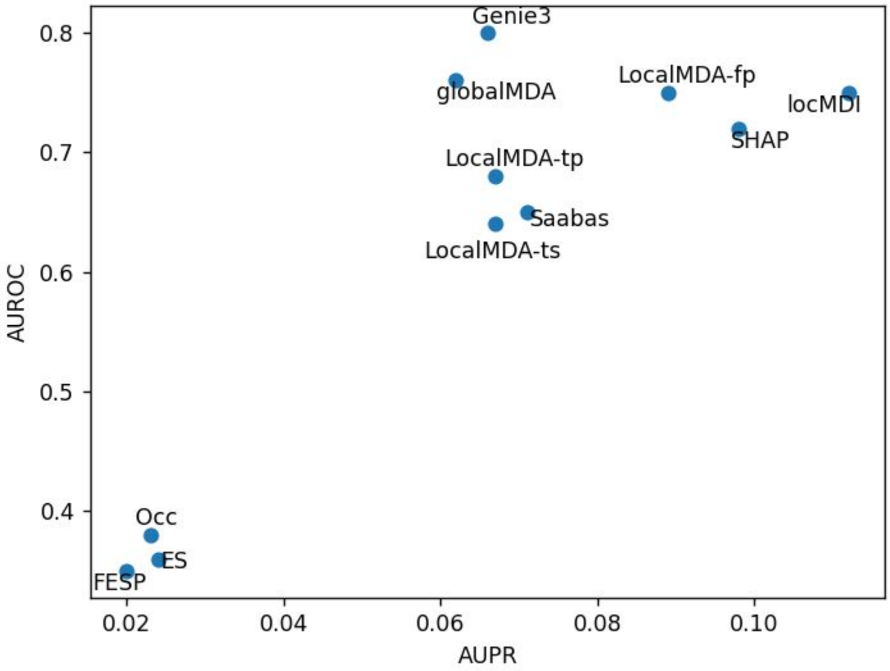


Figure 4.2: *mean of meanAUROC/meanAUPR plot for best normalized local methods and global methods for Dyngen datasets*

The application of local feature importance methods to Dyngen datasets to infer cell-specific GRN provided interesting results.
In terms of AUROC, the normalized `Shap`, localMDI, LocalMDA(forest perturbations) gave close results to the application of global GRN to all cells (tables 4.6, 4.7 and B.1).
The AUROC is a measure of the proportion of existing interactions determined as existing by the method on the proportion of predicted existing interactions among

all the absent ones. The difference shows that either the global methods predict more true positive instances, either they do fewer errors by predicting an existing link when there is not.

In terms of AUPR, the normalized `Shap`, localMDI, LocalMDA(forest perturbations), and base Saabas gave better results (tables 4.8, 4.9 and B.2).
The AUPR is a measure of the proportion of well predicted existing interactions among all predicted existing interactions on the proportion of existing interactions determined as existing by the method. The low scores tell us that the methods predict too many regulatory links that there are. The difference between the local and global methods shows that normalized local methods are more precise at predicting existing interactions.

Local feature importance methods are more precise at predicting interactions between genes at a cell-specific resolution than considering the network predicted globally as the individual prediction for each cell. These methods allow to better distinct existing interactions within each cell than global methods. The overall results are however weak concerning the precision (AUPR) of the methods used, meaning that networks determined are still way too large with respect to the ground truth networks.
The synthetic dataset generation process provide information about results. Each cell-specific GRN is defined from the reference GRN and expression data. Links are not existing in the reference networks will never appear in the cell's networks, only a part of the interactions of the baseline networks will appear in each cell. AUPR scores can therefore be interpreted as a more precise definition of existing links in cells, where the whole network is not expressed.

## 4.3 Cell-type specific datasets

The objective of this section is to benchmark local feature importance methods on a problem concerning GRN inference, especially cell-type specific GRN inference.

As described in section 3.3, with permutations and the use of BoolODE generator, datasets with ten different subtypes were created.

The analysis will focus on determining if by applying local feature importance methods on the whole dataset and averaging over the ten types, better results are obtained than by training global methods on each of the ten cell-type dataset separatedly. The objective is to highlight if the analysis of a bigger dataset allow us to better detect common interactions, while still detecting different ones.

### 4.3.1 Methodology

The cell-types ground-truth regulatory networks were generated by the permutation of the edges of a reference network as described in section 3.3. The datasets obtained contains ten different types defined by 200 single cell gene expression levels each. Three different kinds of reference networks were used, and thus three datasets of ten types each were created.

Global feature importance methods will first be run on each separated datasets and their AUPR and AUROC metrics will be computed. The AUPR and AUROC metrics are deduced from the comparison of the ground truth matrix of interactions of the cell-type network $M$ and matrix of feature importance $M'$ for each method, and for each dataset. The matrix $M$ is of size $g \times (g-1)$. It contains boolean true values 1 for each pair $\{i, j\}$ of row $i$ and column $j$that correspond to a regulation link from $i^{th}$ gene to $j^{th}$ gene, 0 otherwise. However, since our algorithm is not designed to discover self-regulatory links, those were suppressed from the ground truth and the corresponding elements were deleted from the matrix $M$ . The matrix $M'$ is of size $n_{methods} \times g \times (g-1)$. It contains all the feature importance values computed by the $n_{methods}$, with $k^{th}$, $k = (0, ..., g)$ gene as label and all the remaining as features. This matrix thus measures a score for each value and, for each method, can be compared to the matrix $M$ to deduce AUPR and AUROC.

Local feature importance methods will be run on the whole datasets containing all types mixed. The AUPR and AUROC metrics are deduced from the comparison of the ground truth matrix of interactions of the 10 different cell-type networks $M$ and matrix of feature importance $M'$ for each method. The matrix have the same dimensions as earlier, with one more for the 10 different types. In the case of local methods, $M'$ will be constructed by taking the mean of the feature importance values of all cells of a particular type, and thus also has a dimension more that accounts for the 10 types.

### 4.3.2 AUPR and AUROC

AUPR and AUROC metrics are described in the section 1.2.1. In this section, all AUROC and AUPR curves were computed with corresponding functions from `scikit-learn` library.

The AUPR and AUROC metrics are deduced from the comparison of the ground truth matrix of interactions $M$ and matrix of feature importance $M'$ for each method. Global feature importances are compared to the cell-type ground truth regulatory network. The following scores were computed using the `Genie3` algorithm on each of the separated type datasets. Since all subtypes datasets were randomly generated and that it is not interesting to evaluate the method on specific subtypes, the mean of these subtypes scores is taken to evaluate the performance of the three global datasets. The results on each of the global datasets can be found in appendix C

| Dataset | mean |
|---|---|
| meanAUROC | 0.643 |
| meanAUPR | 0.169 |

Table 4.12: *Mean of the meanAUROC/meanAUPR measures over the ten subtypes for Genie3 on the detection of interactions for datasets deduced from three reference networks*

For local methods, the models are trained on the entire dataset (with the ten types mixed) and then the means of the local feature importances of cell of the same types are taken. Then, each averaged feature importances are compared to the cell-type ground truth regulatory network. The following scores were computed using all the local feature importance algorithms on each of the separated type datasets, excepted ES and FESP technique. Indeed, these two methods showed very poor results in the two preceding sections and were abandoned. Since all subtypes datasets were randomly generated and that it is not interesting to evaluate the method on specific subtypes, the mean of these subtypes scores is taken to evaluate the performance. The results on the three different datasets are available in appendix C. The reported scores are the mean across the three datasets.

| meanAUROC \Dataset | mean |
|---|---|
| localMDI | 0.54 |
| `Shap` | 0.56 |
| Saabas | 0.53 |
| LocalMDA - forest perturbations | 0.54 |
| LocalMDA - tree perturbations | 0.57 |
| LocalMDA - tree structure | 0.51 |

Table 4.13: *Mean of the meanAUROC measures over the ten subtypes for local methods on the detection of interactions for datasets deduced from three reference networks*

| meanAUPR \Dataset | mean |
|---|---|
| localMDI | 0.041 |
| `Shap` | 0.044 |
| Saabas | 0.029 |
| LocalMDA - forest perturbations | 0.052 |
| LocalMDA - tree perturbations | 0.052 |
| LocalMDA - tree structure | 0.027 |

Table 4.14: *Mean of the meanAUPR measures over the ten subtypes for local methods on the detection of interactions for datasets deduced from three reference networks*

Comparing the results, it is clear that **global methods applied to each subtype dataset perform better than taking the mean of local methods results applied to the whole dataset for each subtype** (tables 4.12, 4.13, 4.14). AUROC scores for local methods are almost the same as a random classifier (0.5). Even if the AUROC scores for the global method are not high, the difference between the two approaches are significant. AUPR scores are really low for the local methods, meaning they are not precise at all to discern accurate predictions. Results for the global methods are around 4 to 5 times better.

### 4.3.3 Impact of FI values normalisation

A possible area for improvement is to consider feature importance values normalisation. Especially, the normalisation and regularization techniques described in section 2.3 are tested on our problem. The normalisation is applied, for each of the vector of size $g \times (g-1)$ of the matrix $M'$. It means it will result in a normalisation of the interactions in each cell of the dataset. The mean of the local feature importances to determine the vector to be compared with the type-specific ground truth matrix of interactions is thus computed on normalized cell feature importances. Then, to study the global effect of the techniques, the average of the meanAUROC and meanAUPR for the best normalisation compared to the mean of the base scores of the local methods is reported. Complete scores for each normalisation are available in appendix D.

| meanAUROC \normalisation | normalised | base |
|---|---|---|
| localMDI | 0.59 (max) | 0.54 |
| `Shap` | 0.57 (l1, l2, max) | 0.56 |
| Saabas | 0.47 (l1) | 0.53 |
| LocalMDA - forest perturbations | 0.55 (max, minmax) | 0.54 |
| LocalMDA - tree perturbations | 0.54 (l1) | 0.57 |
| LocalMDA - tree structure | 0.49 (minmax) | 0.51 |

Table 4.15: *comparison of base and normalised meanAUROC measures over the ten subtypes for local methods on the detection of interactions for datasets averaged on three reference networks*

| meanAUPR \normalisation | normalised | none |
|---|---|---|
| localMDI | 0.046 (max) | 0.041 |
| `Shap` | 0.0455 (l1) | 0.044 |
| Saabas | 0.027 (max, minmax) | 0.029 |
| LocalMDA - forest perturbations | 0.042 (l1) | 0.052 |
| LocalMDA - tree perturbations | 0.057 (max) | 0.052 |
| LocalMDA - tree structure | 0.04 (l1) | 0.027 |

Table 4.16: *comparison of base and normalized meanAUPR measures over the ten subtypes for local methods on the detection of interactions for datasets averaged on three reference networks*

Impact of feature normalisation on the performance of the model is low. As in 4.2.3, normalisation decreases a lot the results of Saabas. For other methods, results observed are similar to the results without normalisation, and no conclusion can be drawn on the effects of normalisation.

### 4.3.4 Discussion

The methodology proposed in this thesis to infer cell-type specific gene regulatory networks from mixed datasets is inefficient. The application of global methods to datasets with distinct types allows capturing the importance of the features with more precision (figure 4.3).

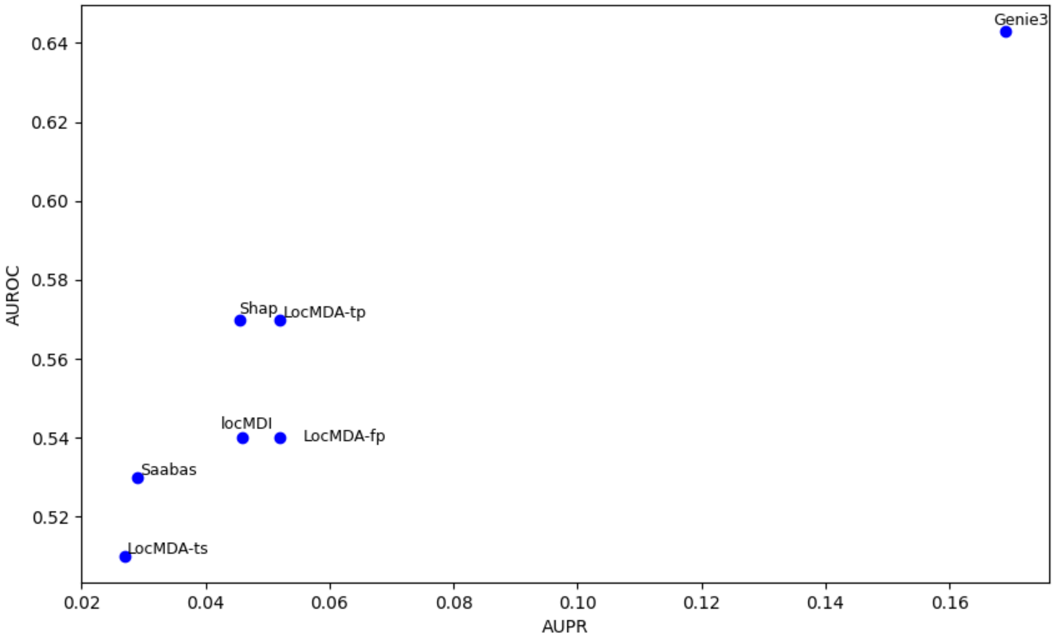


Figure 4.3: *mean of meanAUROC/meanAUPR plot for local methods and global methods on Cell-types datasets*

In particular, the significant differences between AUPR of the global and local methods (tables 4.12 and 4.14) explain an incapacity of the local methodology to effectively distinguish the existing regulations, the methods estimate numerous links as being existing regulations. The AUROC score of local methods is close to the score of a random classifier, demonstrating the inefficiency of the process in classifying existing and non-existing links. The normalisation has no effect on the results.

The local methods take no advantages of running on the whole dataset in the determination of regulatory links of distinct subtypes. The better way to analyse regulatory links is still the application of global methods to each separated dataset.

Results must however be mitigated. Indeed, since neither the local approach, neither the global approach gave strong results on this problem, no strict conclusion can be made on the performance of the methodologies.

## 4.4 CEDAR datasets

The goal of this section is to apply cell-type specific network inference to a real dataset. The dataset used is the CEDAR dataset described in 3.4. This dataset is composed of gene expression data from 27 types of cells of peripheral blood. It contains data from each cell-type for a list of distinct patients, together with a list of 178 genes of interests.

However, this dataset does not provide any ground truth networks. The analysis of the dataset will thus be composed of two steps: the comparison of rankings of the different types, and an analysis of correlation between cells sharing specific genetic markers.

### 4.4.1 Comparison of the rankings for the global and local methods

As no ground truth is available, no metrics earlier used can be applied to this problem.

The methodology to compare the regulatory links found by computing cell-type network inference is hence different. The local methods will be run on the whole dataset with mixed types and average the results of local values for each cell-type (as in 4.3). Then are compared the common and different interactions between each cell-types, for the 10, 20 and 50 interactions with the highest scores.
On the other hand, a global feature importance technique will be applied to each subtype separated dataset. As for the local method, common and different interactions between each cell-types will be computed.
Afterwards, both interactions found by the methods for each cell-type will be compared to see if they find the same interactions of importance.

Only one local method will be applied in this situation. Indeed, the only effective methods earlier tested with practicable computation time for a dataset of this size and number of genes is localMDI.

| Common Interactions between cell-types | #Top 10 | #Top 20 | #Top 50 |
|---|---|---|---|
| Local MDI | 0 | 6 | 40 |
| Genie3 on separated sets | 0 | 0 | 0 |

Table 4.17: *Number of common interactions in the top rankings of interactions between all cell types for an averaged local feature importance method trained on the whole dataset and a global feature importance method trained on separated datasets*

The results in table 4.17 state clearly that while training a model on the whole dataset and taking averaged local feature importance value for cells that share the same type, the gene regulatory links between cell-types found shares a lot of common mechanisms, especially considering larger sets of top scores. On the other hand, no common interactions were found by comparing the top scores of global models trained on each subtype dataset.

Comparing the top 100 interactions between the two methodologies on the 27 types, we found the number of common interactions in both methods rankings in table 4.18. It can be observed that the two methods find common links for each cell-type, but the number is rather low. However, these links found by the two methods should be regulatory links of particular interest, since both different methodologies found them as top 100 dominant among 31 506 interactions. A list of the common links can be found in appendix E.

| Celltypes | Common interactions between methods |
|---|---|
| Classical monocytes | 25 |
| Eosinophils | 12 |
| Granulocytes | 17 |
| ILC | 21 |
| Intermediate moncytes | 23 |
| MAIT | 9 |
| Memory B cells | 18 |
| Memory CD4 | 27 |
| Memory CD8 | 39 |
| Memory Treg | 21 |
| Myeloid DC | 23 |
| Naive B | 12 |
| Naive CD4 | 30 |
| Naive CD8 | 23 |
| Naive Treg | 17 |
| Neutrophils | 20 |
| NK | 13 |
| NKT | 24 |
| Non-classical monocytes | 23 |
| PBMC | 20 |
| Plasmablasts | 23 |
| Plasmacytoid | 19 |
| TCR | 25 |
| Th1 17 | 27 |
| Th1 | 18 |
| Th2 | 17 |
| Th17 | 19 |

Table 4.18: *Number of common interactions in the top 100 ranking of interactions for each cell type between an averaged local feature importance method trained on the whole dataset and a global feature importance method trained on separated datasets*

### 4.4.2 Determination of the genetic marker influence through correlations

Another feature coming with CEDAR dataset is a list of particular genetic markers of interest present on each patient whose sampled cells belong to. These markers either takes the value 0 (absence of the marker) or 1/2 (presence of the marker). It is then possible to use local feature importance techniques on the dataset, and compute the mean of the features scores for each cell of a patient presenting the marker or not.

Next, the Spearman's rank correlation between the averaged feature importance values of the samples with the marker and without is calculated. Highlighting the low correlation scores, it is possible to find genetic markers that have a high influence on gene regulation.

Only one local method will again be applied in this situation. Indeed, the only effective methods earlier tested with practicable computation time for a dataset of this size and number of genes is localMDI.

Comparing a histogram of all correlations for the 206 genetic markers :

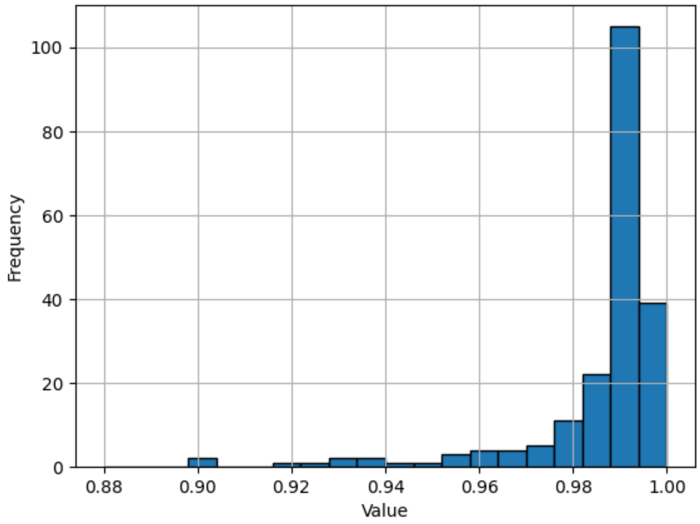


Figure 4.4: *Histogram of correlation values between feature importance values for the absence and presence of each genetic markers*

It can be observed that for a few genetic markers, correlation falls below 0.98. These specific scores indicate genetic markers of interest in the determination of gene regulation, since the feature importance scores are less correlated between patients with and without the marker. In particular, the genetic markers with correlation below 0.95 are

1. chr1:2565210:C:G
2. chr1:67242007:G:A
3. chr1:161499264:G:C
4. chr2:144728756:A:C
5. chr3:47144160:C:G
6. chr4:48264785:C:T
7. chr6:159049210:G:T
8. chr6:166976754:A:G
9. chr7:100924735:G:A
10. chr22:39263768:C:T

### 4.4.3 Discussion

Exploration of CEDAR dataset conducts to interesting results. The evaluation of common rankings (tables 4.17 and 4.18) shows that the use of local feature importance methods on aggregated datasets tends to identify common interactions between types, where the use of global methods on separated datasets do not. However, it is important to confront these results with the underlying biology to confirm whether the common interactions are true common interactions or noise brought to each cell-type networks caused by the common trained model. However, comparing the global methods run on different cell-types dataset and the local methods averaged on the whole dataset for each cell-type, a few interactions can be highlighted. Given the difference in methodology between the techniques, these interactions are links of interest to be investigated.

Concerning the analysis of genetic markers, the application of the method was considered with a comparison of the feature importance values' correlation for cells with and without the marker. These correlation values highlighted a few genetic markers that influence the gene regulation, since correlation between the rankings of the patients presenting or not the marker is lower than expected. These genetic markers should be investigated further by specialist to evaluates their relevance in the gene regulation of blood cells.

# Chapter 5

# Limitations and Conclusion

The objective of this master's thesis was the evaluation of a methodology to infer cell-specific and cell-type specific gene regulatory networks from single cell gene expression data. The approach proposed was to follow the regression technique proposed in GENIE3 algorithm, by using local feature importance scores in place of global scores. To predict the expression of a given gene from the expressions of all other genes (or a subset of candidate transcription factors), a machine learning regression model is trained from a dataset of single-cell measurements. From the local importance scores that the machine learning model derives for each cell (or cell type) in the dataset, a cell-specific or cell-type specific network can then be derived.

To this extent, a selection of local feature importance attribution methods have been investigated. The methods evaluated were `Shap`, local mean decrease of impurity (localMDI), Saabas, fair-equivalent-symmetric-perturbation values (FESP), and equal-surplus values (ES). Three different implementations of local mean decrease of accuracy (local MDA) were also completed and examined.

At first, the evaluation of the local methods on a simple regression problem was performed. The objective was to assess the quality of the deduced importance scores on the Friedman 1 dataset, whose outputs are derived from a small subset of features. The obtained results for each sample by local methods were aggregated to be compared to the findings of global feature importance methods (Genie3, globalMDA). It was shown that the average of local scores managed to match the AUROC and AUPR scores of global methods for four different local techniques (`Shap`, localMDI, and the two implementations of localMDA based on perturbations of the samples). These algorithms are thus valid for the characterisation of more complex problems. It was also shown by the analysis of Spearman's rank correlation that the three implementations of localMDA were sharing similarities, as well as the four methods performing well on the dataset.

Next, the local feature importance techniques were tested on the cell-specific network inference problem. The datasets of interest were synthetic datasets generated using the Dyngen simulator, whose cell-specific ground truth networks were known. An analysis of the local and global methods showed that consideration of the rankings made by global methods as cell-specific networks provided better results. However, after application of normalisation at the cell-specific genes level of the importance scores, three methods outperformed the global rankings methods. `Shap`, localMDI and the implementation of localMDA based on perturbations at the level of the forest gave similar AUROC results, but better AUPR score. FESP and ES showed terrible results and were abandoned. Local methods allow a thus deeper characterization of cell-specific regulation mechanisms.

The cell-type specific network inference challenge was then assessed using the local feature importance algorithms. To this extent, a synthetic dataset with permutations to mimic cell-types networks was created. Cell's feature importance scores computed with local techniques were aggregated following each cell's type, and compared to global networks. These networks were found by the analysis of Genie3 on each separated cell-type dataset. The AUPR and AUROC scores of all local approach were lower than the scores of global methods. The approach proposed in this case is not working, the common interactions detected by the model trained on the whole datasets are too disruptive. However, results of both local and global methods were poor.

Finally, a real dataset (CEDAR) of 27 different cell-types issued from peripheral blood was investigated. As no ground-truth networks were available, the analysis summarised in comparing the top ranked interactions by the local mean decrease of impurity method and the Genie3 algorithm applied on cell-types separately. The results showed that the interactions found by localMDI were shared between types, while not at all for the Genie3 approach. However, setting side by side the rankings of the two approaches for each cell-type, a list of common interactions for both methodologies was deduced. These interactions are thus regulation links of interests to be explored.
Another look was brought to the association of genetic markers to each cell of the real dataset, CEDAR. Comparison was made between a mean of localMDI scores for

the cell of patients presenting a specific marker or not. Analysing the Spearman's correlation between both permitted to highlight low correlation scores inducing a high interference of the marker in the gene regulation process. A list of ten markers was established to be interesting.

The investigation of cell-specific and cell-type specific gene regulatory networks however strongly depends on the synthetic datasets and their simulators' reliability. Indeed, as no ground-truth is available for the dataset with real measurements, it can not be concluded that the results are completely trustworthy. An area of improvement would hence be the evaluation of ground-truth real datasets as they will be available.

Furthermore, the advances in deep learning field could be an avenue for improving. Models as RandomForest and other tree-based methods are being more and more outperformed by the evolution of deep learning-base models. A development of local feature importance attribution algorithms on these models could allow in-depth understanding of some process not captured by tree models.

Overall, even if the results on cell-type and cell-specific GRN inference were mixed, the field holds potential for further research and innovation. By continuously exploring new technologies, ground-truth measurements, importance features applied to modern learning models, future advances could be brought in the field.

# Bibliography

[1] Vân Anh Huynh-Thu et al. "Inferring regulatory networks from expression data using tree-based methods". In: *PLoS ONE* 5.9 (Sept. 2010). DOI: 10.1371/journal.pone.0012776.

[2] Pierre Geurts. *An introduction to  Machine Learning*. URL: https://people.montefiore.uliege.be/lwh/AIA/.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of Statistical Learning, second edition: Data Mining, Inference, and prediction*. Springer, 2009.

[4] Gilles Louppe. "Understanding Random Forests: From Theory to Practice". English. PhD thesis. ULiège - Université de Liège, October 2014. URL: http://github.com/glouppe/phd-thesis.

[5] Aurélien Géron. *Hands-on machine learning with scikit-learn and tensorflow: Concepts, tools, and techniques to build Intelligent Systems*. O'Reilly Media, 2017.

[6] Pierre Geurts and Louis Wehenkel. *Bias/variance trade-off, model assessment and model selection*. URL: https://people.montefiore.uliege.be/lwh/AIA/.

[7] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Chapman  Hall/CRC, 2000.

[8] Lisa A. Urry et al. *Campbell Biology + Masteringbiology with Etext Access Card*. Pearson College Div, 2016.

[9] Aditya Pratapa et al. "Benchmarking algorithms for Gene Regulatory Network inference from single-cell transcriptomic data". In: *Nature Methods* 17.2 (Jan. 2020), pp. 147–154. DOI: 10.1038/s41592-019-0690-6.

[10] Todorov, Helena and Cannoodt, Robrecht and Saelens, Wouter and Saeys, Yvan. "Network inference from single-cell transcriptomic data". eng. In: *Gene regulatory networks : methods and protocols*. Vol. 1883. Methods in Molecular Biology. Humana Press, 2019, 235–249. ISBN: 9781493988815. URL: %7Bhttp://doi.org/10.1007/978-1-4939-8882-2_10%7D.

[11] Marieke Lydia Kuijjer et al. "Estimating sample-specific regulatory networks". In: *iScience* 14 (Apr. 2019), pp. 226–240. DOI: 10.1016/j.isci.2019.03.021.

[12] Sara Aibar et al. "Scenic: Single-cell regulatory network inference and clustering". In: *Nature Methods* 14.11 (Oct. 2017), pp. 1083–1086. DOI: 10.1038/nmeth.4463.

[13] Philipp Keyl et al. "Single-cell gene regulatory network prediction by explainable AI". In: *Nucleic Acids Research* 51.4 (Jan. 2023). DOI: 10.1093/nar/gkac1212.

[14] Leo Breiman et al. *Classification and regression trees*. Chapman  Hall/CRC, 2017.

[15] Antonio Sutera et al. *From global to local MDI variable importances for random forests and when they are Shapley values*. 2021. arXiv: 2111.02218 [stat.ML]. URL: https://arxiv.org/abs/2111.02218.

[16] Christoph Molnar. *Interpretable machine learning: A guide for making Black Box models explainable*. Christoph Molnar, 2022.

[17] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously". In: *Journal of Machine Learning Research* 20.177 (2019), pp. 1–81. URL: http://jmlr.org/papers/v20/18-760.html.

[18] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874.

[19] Hima Lakkaraju, Julius Adebayo, and Sameer Singh. *Explaining Machine Learning Predictions: State-of-the-art, Challenges, Opportunities*. URL: https://explainml-tutorial.github.io/neurips20.

[20] Christoph Molnar. *Interpreting machine learning models with Shap: A Guide with python examples and theory on Shapley Values*. ChistophMolnar c/o MUCBOOK, Heidi Seibold, 2023.

[21] L. S. Shapley. "17. A value for n-person games". In: *Contributions to the Theory of Games (AM-28), Volume II* (Dec. 1953), pp. 307–318. DOI: 10.1515/9781400881970-018.

[22] Shap. *Shap/SHAP: A game theoretic approach to explain the output of any machine learning model*. URL: https://github.com/shap/shap.

[23] Sujit R. Tangadpalliwar on December 24 et al. Oct. 2014. URL: https://blog.datadive.net/interpreting-random-forests/.

[24] Shap. *Why is saabas considered as an approximation for shap? · issue 323 · Shap/Shap*. URL: https://github.com/shap/shap/issues/323.

[25] Asutera. *ASUTERA/local-MDI-importance: Repository for the paper "from global to local MDI variable importances for random forests and when they are Shapley values"*. URL: https://github.com/asutera/Local-MDI-importance/tree/main.

[26] Pierre-François Weyders. "Local permutation importances for random forests". PhD thesis. ULiège, 2021.

[27] scikit learn developers. URL: https://scikit-learn.org/stable/auto_examples/tree/plot_unveil_tree_structure.html.

[28] Charles Condevaux, Sébastien Harispe, and Stéphane Mussard. "Fair and efficient alternatives to shapley-based attribution methods". In: *Lecture Notes in Computer Science* (2023), pp. 309–324. DOI: 10.1007/978-3-031-26387-3_19.

[29] Celestin Chameni and Nicolas Andjiga. "Linear, efficient and symmetric values for TU-games". In: *Economics Bulletin* 3 (Dec. 2008), pp. 1–10.

[30] T. S. Driessen and Y. Funaki. "Coincidence of and collinearity between game Theoretic Solutions". In: *Operations-Research-Spektrum* 13.1 (Mar. 1991), pp. 15–30. DOI: 10.1007/bf01719767.

[31] Ccdv-Ai. *CCDV-ai/FESP$_E$S*. URL: https://github.com/ccdv-ai/fesp_es/tree/main.

[32] Jerome H. Friedman. "Multivariate Adaptive Regression Splines". In: *The Annals of Statistics* 19.1 (1991), pp. 1–67. DOI: 10.1214/aos/1176347963. URL: https://doi.org/10.1214/aos/1176347963.

[33] scikit learn developers. URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman1.html#make-friedman1.

[34] Robrecht Cannoodt et al. "Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells". In: *Nature Communications* 12.1 (June 2021). DOI: 10.1038/s41467-021-24152-2.

[35] Thomas Schaffter, Daniel Marbach, and Dario Floreano. "GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods". In: *Bioinformatics* 27.16 (June 2011), pp. 2263–2270. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr373. eprint: https://academic.oup.com/bioinformatics/article-pdf/27/16/2263/48863257/bioinformatics_27_16_2263.pdf. URL: https://doi.org/10.1093/bioinformatics/btr373.

# Appendix A

# Dyngen detailed meanAUPR and meanAUROC values

| Local Methods \ Dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Local MDI | **0.755** | 0.60 | 0.63 | **0.64** | 0.78 | 0.79 | 0.81 |
| Shap | 0.70 | **0.62** | 0.47 | 0.65 | **0.83** | **0.84** | **0.85** |
| Saabas | 0.69 | 0.52 | **0.62** | 0.61 | 0.73 | 0.73 | 0.75 |
| LocalMDA - forest perturbations | 0.70 | 0.59 | 0.61 | 0.64 | 0.81 | 0.83 | 0.83 |
| LocalMDA - tree perturbations | 0.71 | 0.59 | 0.52 | 0.63 | 0.70 | 0.72 | 0.72 |
| LocalMDA - tree structure | 0.68 | 0.57 | 0.51 | 0.62 | 0.67 | 0.69 | 0.67 |
| FESP | 0.33 | 0.45 | 0.34 | 0.40 | 0.22 | 0.26 | 0.18 |
| ES | 0.26 | 0.42 | 0.38 | 0.37 | 0.23 | 0.27 | 0.19 |

Table A.1: *meanAUROC score for local feature importance methods for the Dyngen datasets 1-7*

| Local Methods \ Dataset | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Local MDI | 0.75 | 0.69 | 0.60 | **0.86** | 0.65 | 0.45 | **0.75** |
| Shap | **0.77** | **0.71** | 0.62 | 0.85 | **0.72** | 0.49 | 0.72 |
| Saabas | 0.63 | 0.61 | **0.66** | 0.78 | 0.67 | **0.53** | 0.57 |
| LocalMDA - forest perturbations | 0.76 | 0.69 | 0.60 | **0.86** | 0.68 | 0.47 | 0.67 |
| LocalMDA - tree perturbations | 0.68 | 0.68 | 0.55 | 0.83 | 0.59 | 0.44 | 0.67 |
| LocalMDA - tree structure | 0.67 | 0.66 | 0.53 | 0.80 | 0.57 | 0.45 | 0.63 |
| FESP | / | / | 0.37 | / | 0.28 | 0.45 | 0.44 |
| ES | / | / | 0.34 | / | 0.30 | 0.41 | 0.45 |

Table A.2: *meanAUROC score for local feature importance methods for the Dyngen datasets 7-14*

| Global methods \dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Global MDA | **0.81** | **0.69** | 0.71 | 0.71 | 0.91 | 0.88 | 0.93 |
| Genie 3 | 0.78 | 0.63 | **0.81** | **0.74** | **0.92** | **0.92** | **0.93** |

Table A.3: *meanAUROC score for global feature importance methods for the Dyngen datasets 1-7*

| Global methods \dataset | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Global MDA | **0.85** | 0.69 | 0.72 | 0.78 | 0.77 | 0.54 | 0.68 |
| Genie 3 | **0.85** | **0.75** | **0.80** | **0.89** | **0.82** | **0.61** | **0.78** |

Table A.4: *meanAUROC score for global feature importance methods for the Dyngen datasets 7-14*

| Local Methods \ Dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Local MDI | 0.046 | 0.029 | 0.048 | 0.058 | 0.054 | 0.060 | 0.065 |
| Shap | 0.057 | **0.036** | 0.018 | 0.054 | 0.077 | 0.069 | 0.066 |
| Saabas | **0.076** | 0.028 | **0.069** | **0.062** | **0.081** | **0.085** | **0.089** |
| LocalMDA - forest perturbations | 0.045 | 0.034 | 0.045 | 0.047 | 0.058 | 0.047 | 0.056 |
| LocalMDA - tree perturbations | 0.029 | 0.027 | 0.029 | 0.033 | 0.026 | 0.022 | 0.024 |
| LocalMDA - tree structure | 0.023 | 0.023 | 0.026 | 0.027 | 0.017 | 0.018 | 0.018 |
| FESP | 0.015 | 0.019 | 0.025 | 0.020 | 0.007 | 0.010 | 0.008 |
| ES | 0.013 | 0.015 | 0.020 | 0.017 | 0.011 | 0.010 | 0.010 |
| Occlusion | 0.028 | 0.024 | 0.021 | 0.026 | 0.012 | 0.017 | 0.011 |

Table A.5: *meanAUPR score for local feature importance methods for the Dyngen datasets 1-7*

| Local Methods \ Dataset | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Local MDI | 0.037 | 0.062 | 0.074 | 0.049 | 0.059 | 0.036 | 0.060 |
| Shap | 0.054 | 0.072 | 0.074 | 0.094 | **0.071** | 0.037 | 0.063 |
| Saabas | **0.060** | **0.085** | **0.088** | **0.097** | 0.068 | 0.039 | **0.068** |
| LocalMDA - forest perturbations | 0.044 | 0.058 | 0.052 | 0.081 | 0.055 | 0.035 | 0.057 |
| LocalMDA - tree perturbations | 0.025 | 0.043 | 0.032 | 0.041 | 0.025 | 0.029 | 0.059 |
| LocalMDA - tree structure | 0.021 | 0.040 | 0.027 | 0.024 | 0.021 | 0.029 | 0.048 |
| FESP | / | / | 0.024 | / | 0.013 | 0.030 | 0.044 |
| ES | / | / | 0.020 | / | 0.022 | 0.028 | 0.031 |
| Occlusion | / | / | 0.042 | / | 0.022 | **0.043** | 0.036 |

Table A.6: *meanAUPR score for local feature importance methods for the Dyngen datasets 7-14*

| Global methods \dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Global MDA | **0.091** | **0.040** | 0.058 | **0.054** | **0.080** | **0.107** | **0.109** |
| Genie 3 | 0.043 | 0.031 | **0.064** | 0.043 | 0.068 | 0.087 | **0.109** |

Table A.7: *meanAUPR score for global feature importance methods for the Dyngen datasets 1-7*

| Global methods \dataset | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Global MDA | 0.039 | 0.063 | 0.060 | 0.017 | 0.078 | 0.034 | 0.042 |
| Genie 3 | **0.053** | **0.067** | **0.102** | **0.038** | **0.127** | **0.043** | **0.049** |

Table A.8: *meanAUPR score for global feature importance methods for the Dyngen datasets 7-14*

# Appendix B

# Normalised feature importances AUROC and AUPR scores for dyngen datasets

| Local methods \normalisation | l1 | l2 | max | min-max | base |
|---|---|---|---|---|---|
| LocalMDI | 0.74 | 0.75 | 0.75 | 0.72 | 0.69 |
| Shap | 0.72 | 0.72 | 0.72 | 0.72 | 0.7 |
| Saabas | 0.65 | 0.65 | 0.65 | 0.64 | 0.65 |
| LocalMDA - forest perturbations | 0.75 | 0.75 | 0.75 | 0.75 | 0.69 |
| LocalMDA - tree perturbations | 0.70 | 0.70 | 0.68 | 0.68 | 0.64 |
| LocalMDA - tree structure | 0.68 | 0.68 | 0.64 | 0.64 | 0.62 |
| FESP | 0.34 | 0.34 | 0.35 | 0.31 | 0.34 |
| ES | 0.35 | 0.35 | 0.35 | 0.36 | 0.33 |
| Occlusion | 0.364 | 0.353 | 0.368 | 0.381 | 0.341 |

Table B.1: *mean of meanAUROC score for normalised local feature importance methods for the Dyngen datasets*

| Local methods \normalisation | l1 | l2 | max | min-max | base |
|---|---|---|---|---|---|
| LocalMDI | 0.066 | 0.07 | 0.112 | 0.049 | 0.048 |
| Shap | 0.06 | 0.06 | 0.098 | 0.0605 | 0.060 |
| Saabas | 0.054 | 0.07 | 0.031 | 0.031 | 0.071 |
| LocalMDA - forest perturbations | 0.050 | 0.050 | 0.089 | 0.076 | 0.051 |
| LocalMDA - tree perturbations | 0.0415 | 0.0414 | 0.067 | 0.036 | 0.031 |
| LocalMDA - tree structure | 0.04 | 0.04 | 0.067 | 0.032 | 0.025 |
| FESP | 0.017 | 0.019 | 0.020 | 0.016 | 0.019 |
| ES | 0.018 | 0.019 | 0.024 | 0.024 | 0.024 |
| Occlusion | 0.024 | 0.024 | 0.024 | 0.023 | 0.026 |

Table B.2: *mean of meanAUPR score for normalised local feature importance methods for the Dyngen datasets*

# Appendix C

# Cell-types datasets detailed AUPR and AUROC values

| Dataset | 1 | 2 | 3 | mean |
|---|---|---|---|---|
| meanAUROC | 0.61 | 0.645 | 0.675 | 0.643 |
| meanAUPR | 0.156 | 0.127 | 0.225 | 0.169 |

Table C.1: *meanAUROC/meanAUPR measures over the 10 sub-types for Genie3 on the detection of interactions for datasets deduced from reference networks 1, 2 and 3*

| meanAUROC \Dataset | 1 | 2 | 3 | mean |
|---|---|---|---|---|
| localMDI | 0.52 | 0.46 | 0.64 | 0.54 |
| `Shap` | 0.56 | 0.47 | 0.64 | 0.56 |
| Saabas | 0.55 | 0.56 | 0.48 | 0.53 |
| LocalMDA - forest perturbations | 0.57 | 0.47 | 0.58 | 0.54 |
| LocalMDA - tree perturbations | 0.58 | 0.55 | 0.58 | 0.57 |
| LocalMDA - tree structure | 0.55 | 0.47 | 0.52 | 0.51 |

Table C.2: *meanAUROC measures over the 10 sub-types for local methods on the detection of interactions for datasets deduced from reference networks 1, 2 and 3*

| meanAUPR \Dataset | 1 | 2 | 3 | mean |
|---|---|---|---|---|
| localMDI | 0.034 | 0.042 | 0.047 | 0.041 |
| Shap | 0.04 | 0.044 | 0.048 | 0.044 |
| Saabas | 0.029 | 0.035 | 0.024 | 0.029 |
| LocalMDA - forest perturbations | 0.036 | 0.04 | 0.07 | 0.052 |
| LocalMDA - tree perturbations | 0.04 | 0.047 | 0.07 | 0.052 |
| LocalMDA - tree structure | 0.028 | 0.032 | 0.023 | 0.027 |

Table C.3: *meanAUPR measures over the 10 sub-types for local methods on the detection of interactions for datasets deduced from reference networks 1, 2 and 3*

# Appendix D

# Normalised feature importances AUROC and AUPR scores for cell-type dataset

| meanAUROC \normalisation | l1 | l2 | max | minmax | none |
|---|---|---|---|---|---|
| localMDI | 0.56 | 0.58 | 0.59 | 0.52 | 0.54 |
| Shap | 0.57 | 0.57 | 0.57 | 0.56 | 0.56 |
| Saabas | 0.47 | 0.45 | 0.45 | 0.43 | 0.53 |
| LocalMDA - forest perturbations | 0.52 | 0.52 | 0.55 | 0.55 | 0.54 |
| LocalMDA - tree perturbations | 0.54 | 0.52 | 0.52 | 0.52 | 0.57 |
| LocalMDA - tree structure | 0.45 | 0.45 | 0.46 | 0.49 | 0.51 |

Table D.1: *comparison of base and normalized meanAUROC measures over the 10 sub-types for local methods on the detection of interactions for datasets averaged on reference networks 1, 2 and 3*

| meanAUPR \normalisation | l1 | l2 | max | minmax | none |
|---|---|---|---|---|---|
| localMDI | 0.045 | 0.043 | 0.046 | 0.045 | 0.041 |
| Shap | 0.0455 | 0.042 | 0.043 | 0.043 | 0.044 |
| Saabas | 0.025 | 0.025 | 0.027 | 0.027 | 0.029 |
| LocalMDA - forest perturbations | 0.042 | 0.04 | 0.041 | 0.041 | 0.052 |
| LocalMDA - tree perturbations | 0.052 | 0.049 | 0.057 | 0.047 | 0.052 |
| LocalMDA - tree structure | 0.04 | 0.039 | 0.033 | 0.036 | 0.027 |

Table D.2: *comparison of base and normalized meanAUPR measures over the 10 sub-types for local methods on the detection of interactions for datasets averaged on reference networks 1, 2 and 3*

# Appendix E

# Common links between localMDI and global Genie3 methods on 27 types of CEDAR dataset

Here is a description of common links between the top 100 regulation links of both methods used in section 4.4. The set numbers are corresponding to the order of cell types defined in 4.4. Common elements between set 0 of the local method (MDI) and set 0 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000142444 -> ENSG00000161179
3. Gene pair : ENSG00000198355 -> ENSG00000151414
4. Gene pair : ENSG00000169203 -> ENSG00000183604
5. Gene pair : ENSG00000112977 -> ENSG00000198355
6. Gene pair : ENSG00000241945 -> ENSG00000160221
7. Gene pair : ENSG00000161179 -> ENSG00000142444
8. Gene pair : ENSG00000268575 -> ENSG00000215790
9. Gene pair : ENSG00000009790 -> ENSG00000102901
10. Gene pair : ENSG00000198355 -> ENSG00000112977
11. Gene pair : ENSG00000149485 -> ENSG00000134824
12. Gene pair : ENSG00000172057 -> ENSG00000073605

13. Gene pair : ENSG00000130592 -> ENSG00000102879
14. Gene pair : ENSG00000102901 -> ENSG00000157873
15. Gene pair : ENSG00000160221 -> ENSG00000241945
16. Gene pair : ENSG00000198355 -> ENSG00000152518
17. Gene pair : ENSG00000134824 -> ENSG00000149485
18. Gene pair : ENSG00000073605 -> ENSG00000172057
19. Gene pair : ENSG00000102901 -> ENSG00000009790
20. Gene pair : ENSG00000152518 -> ENSG00000198355
21. Gene pair : ENSG00000102901 -> ENSG00000143224
22. Gene pair : ENSG00000215790 -> ENSG00000268575
23. Gene pair : ENSG00000183604 -> ENSG00000169203
24. Gene pair : ENSG00000102879 -> ENSG00000130592
25. Gene pair : ENSG00000176340 -> ENSG00000184076

Number of common elements: 25

Common elements between set 1 of the local method (MDI) and set 1 of the global method:

1. Gene pair : ENSG00000172057 -> ENSG00000073605
2. Gene pair : ENSG00000152518 -> ENSG00000168286
3. Gene pair : ENSG00000241945 -> ENSG00000160221
4. Gene pair : ENSG00000168286 -> ENSG00000152518
5. Gene pair : ENSG00000172543 -> ENSG00000172500
6. Gene pair : ENSG00000144381 -> ENSG00000115484
7. Gene pair : ENSG00000160221 -> ENSG00000241945
8. Gene pair : ENSG00000115484 -> ENSG00000144381
9. Gene pair : ENSG00000198355 -> ENSG00000152518

10. Gene pair : ENSG00000134824 -> ENSG00000149485
11. Gene pair : ENSG00000073605 -> ENSG00000172057
12. Gene pair : ENSG00000149485 -> ENSG00000134824

Number of common elements: 12

Common elements between set 2 of the local method (MDI) and set 2 of the global method:

1. Gene pair : ENSG00000189339 -> ENSG00000248333
2. Gene pair : ENSG00000241945 -> ENSG00000160221
3. Gene pair : ENSG00000161179 -> ENSG00000142444
4. Gene pair : ENSG00000142444 -> ENSG00000161179
5. Gene pair : ENSG00000248333 -> ENSG00000189339
6. Gene pair : ENSG00000136240 -> ENSG00000142444
7. Gene pair : ENSG00000142444 -> ENSG00000090238
8. Gene pair : ENSG00000160221 -> ENSG00000241945
9. Gene pair : ENSG00000183604 -> ENSG00000169203
10. Gene pair : ENSG00000142444 -> ENSG00000136240
11. Gene pair : ENSG00000141367 -> ENSG00000005844
12. Gene pair : ENSG00000090238 -> ENSG00000142444
13. Gene pair : ENSG00000268575 -> ENSG00000215790
14. Gene pair : ENSG00000169203 -> ENSG00000183604
15. Gene pair : ENSG00000134824 -> ENSG00000149485
16. Gene pair : ENSG00000215790 -> ENSG00000268575
17. Gene pair : ENSG00000149485 -> ENSG00000134824

Number of common elements: 17

Common elements between set 3 of the local method (MDI) and set 3 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000168286 -> ENSG00000152518
3. Gene pair : ENSG00000090238 -> ENSG00000142444
4. Gene pair : ENSG00000176340 -> ENSG00000184076
5. Gene pair : ENSG00000169203 -> ENSG00000183604
6. Gene pair : ENSG00000169203 -> ENSG00000205534
7. Gene pair : ENSG00000081154 -> ENSG00000117500
8. Gene pair : ENSG00000241945 -> ENSG00000160221
9. Gene pair : ENSG00000268575 -> ENSG00000215790
10. Gene pair : ENSG00000149485 -> ENSG00000134824
11. Gene pair : ENSG00000172057 -> ENSG00000073605
12. Gene pair : ENSG00000189339 -> ENSG00000248333
13. Gene pair : ENSG00000160221 -> ENSG00000241945
14. Gene pair : ENSG00000198355 -> ENSG00000152518
15. Gene pair : ENSG00000134824 -> ENSG00000149485
16. Gene pair : ENSG00000073605 -> ENSG00000172057
17. Gene pair : ENSG00000205534 -> ENSG00000169203
18. Gene pair : ENSG00000152518 -> ENSG00000198355
19. Gene pair : ENSG00000215790 -> ENSG00000268575
20. Gene pair : ENSG00000183604 -> ENSG00000169203
21. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 21

Common elements between set 4 of the local method (MDI) and set 4 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340

2. Gene pair : ENSG00000142444 -> ENSG00000136240
3. Gene pair : ENSG00000176986 -> ENSG00000196305
4. Gene pair : ENSG00000176340 -> ENSG00000184076
5. Gene pair : ENSG00000141367 -> ENSG00000176986
6. Gene pair : ENSG00000169203 -> ENSG00000183604
7. Gene pair : ENSG00000102879 -> ENSG00000164054
8. Gene pair : ENSG00000241945 -> ENSG00000160221
9. Gene pair : ENSG00000149485 -> ENSG00000134824
10. Gene pair : ENSG00000172057 -> ENSG00000073605
11. Gene pair : ENSG00000136240 -> ENSG00000142444
12. Gene pair : ENSG00000160221 -> ENSG00000241945
13. Gene pair : ENSG00000115484 -> ENSG00000144381
14. Gene pair : ENSG00000134824 -> ENSG00000149485
15. Gene pair : ENSG00000073605 -> ENSG00000172057
16. Gene pair : ENSG00000215790 -> ENSG00000189339
17. Gene pair : ENSG00000164054 -> ENSG00000206503
18. Gene pair : ENSG00000172500 -> ENSG00000172543
19. Gene pair : ENSG00000176986 -> ENSG00000141367
20. Gene pair : ENSG00000172543 -> ENSG00000172500
21. Gene pair : ENSG00000144381 -> ENSG00000115484
22. Gene pair : ENSG00000183604 -> ENSG00000169203
23. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 23

Common elements between set 5 of the local method (MDI) and set 5 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000241945 -> ENSG00000160221
3. Gene pair : ENSG00000115484 -> ENSG00000150753
4. Gene pair : ENSG00000189339 -> ENSG00000215790
5. Gene pair : ENSG00000160221 -> ENSG00000241945
6. Gene pair : ENSG00000134824 -> ENSG00000149485
7. Gene pair : ENSG00000176340 -> ENSG00000184076
8. Gene pair : ENSG00000215790 -> ENSG00000189339
9. Gene pair : ENSG00000149485 -> ENSG00000134824

Number of common elements: 9

Common elements between set 6 of the local method (MDI) and set 6 of the global method:

1. Gene pair : ENSG00000168286 -> ENSG00000198355
2. Gene pair : ENSG00000198355 -> ENSG00000168286
3. Gene pair : ENSG00000184076 -> ENSG00000176340
4. Gene pair : ENSG00000152518 -> ENSG00000168286
5. Gene pair : ENSG00000152518 -> ENSG00000198355
6. Gene pair : ENSG00000241945 -> ENSG00000160221
7. Gene pair : ENSG00000168286 -> ENSG00000152518
8. Gene pair : ENSG00000136240 -> ENSG00000142444
9. Gene pair : ENSG00000149485 -> ENSG00000134824
10. Gene pair : ENSG00000160221 -> ENSG00000241945
11. Gene pair : ENSG00000268575 -> ENSG00000215790
12. Gene pair : ENSG00000142444 -> ENSG00000136240
13. Gene pair : ENSG00000010256 -> ENSG00000102879

14. Gene pair : ENSG00000198355 -> ENSG00000152518

15. Gene pair : ENSG00000134824 -> ENSG00000149485

16. Gene pair : ENSG00000215790 -> ENSG00000268575

17. Gene pair : ENSG00000215790 -> ENSG00000189339

18. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 18
Common elements between set 7 of the local method (MDI) and set 7 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340

2. Gene pair : ENSG00000142444 -> ENSG00000161179

3. Gene pair : ENSG00000142444 -> ENSG00000136240

4. Gene pair : ENSG00000176340 -> ENSG00000184076

5. Gene pair : ENSG00000169203 -> ENSG00000183604

6. Gene pair : ENSG00000144381 -> ENSG00000150753

7. Gene pair : ENSG00000102879 -> ENSG00000164054

8. Gene pair : ENSG00000241945 -> ENSG00000160221

9. Gene pair : ENSG00000161179 -> ENSG00000142444

10. Gene pair : ENSG00000115484 -> ENSG00000150753

11. Gene pair : ENSG00000148090 -> ENSG00000161179

12. Gene pair : ENSG00000161179 -> ENSG00000148090

13. Gene pair : ENSG00000149485 -> ENSG00000134824

14. Gene pair : ENSG00000172046 -> ENSG00000149923

15. Gene pair : ENSG00000172057 -> ENSG00000073605

16. Gene pair : ENSG00000149923 -> ENSG00000172046

17. Gene pair : ENSG00000136240 -> ENSG00000142444

18. Gene pair : ENSG00000160221 -> ENSG00000241945
19. Gene pair : ENSG00000134824 -> ENSG00000149485
20. Gene pair : ENSG00000073605 -> ENSG00000172057
21. Gene pair : ENSG00000215790 -> ENSG00000189339
22. Gene pair : ENSG00000150753 -> ENSG00000115484
23. Gene pair : ENSG00000205534 -> ENSG00000169203
24. Gene pair : ENSG00000135926 -> ENSG00000261338
25. Gene pair : ENSG00000164054 -> ENSG00000102879
26. Gene pair : ENSG00000183604 -> ENSG00000169203
27. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 27

Common elements between set 8 of the local method (MDI) and set 8 of the global method:

1. Gene pair : ENSG00000149923 -> ENSG00000068831
2. Gene pair : ENSG00000184076 -> ENSG00000176340
3. Gene pair : ENSG00000142444 -> ENSG00000161179
4. Gene pair : ENSG00000142444 -> ENSG00000090238
5. Gene pair : ENSG00000142444 -> ENSG00000136240
6. Gene pair : ENSG00000068831 -> ENSG00000149923
7. Gene pair : ENSG00000090238 -> ENSG00000142444
8. Gene pair : ENSG00000169203 -> ENSG00000183604
9. Gene pair : ENSG00000112977 -> ENSG00000198355
10. Gene pair : ENSG00000144381 -> ENSG00000150753
11. Gene pair : ENSG00000241945 -> ENSG00000160221
12. Gene pair : ENSG00000115232 -> ENSG00000081237

13. Gene pair : ENSG00000161179 -> ENSG00000142444

14. Gene pair : ENSG00000115484 -> ENSG00000150753

15. Gene pair : ENSG00000268575 -> ENSG00000215790

16. Gene pair : ENSG00000148090 -> ENSG00000161179

17. Gene pair : ENSG00000009790 -> ENSG00000102901

18. Gene pair : ENSG00000198355 -> ENSG00000112977

19. Gene pair : ENSG00000161179 -> ENSG00000148090

20. Gene pair : ENSG00000149485 -> ENSG00000134824

21. Gene pair : ENSG00000172046 -> ENSG00000149923

22. Gene pair : ENSG00000196502 -> ENSG00000178952

23. Gene pair : ENSG00000149923 -> ENSG00000172046

24. Gene pair : ENSG00000136240 -> ENSG00000142444

25. Gene pair : ENSG00000160221 -> ENSG00000241945

26. Gene pair : ENSG00000010256 -> ENSG00000102879

27. Gene pair : ENSG00000115484 -> ENSG00000144381

28. Gene pair : ENSG00000134824 -> ENSG00000149485

29. Gene pair : ENSG00000215790 -> ENSG00000189339

30. Gene pair : ENSG00000150753 -> ENSG00000115484

31. Gene pair : ENSG00000102879 -> ENSG00000010256

32. Gene pair : ENSG00000151151 -> ENSG00000096968

33. Gene pair : ENSG00000152518 -> ENSG00000198355

34. Gene pair : ENSG00000096968 -> ENSG00000151151

35. Gene pair : ENSG00000178952 -> ENSG00000196502

36. Gene pair : ENSG00000144381 -> ENSG00000115484

37. Gene pair : ENSG00000183604 -> ENSG00000169203

38. Gene pair : ENSG00000150753 -> ENSG00000144381

39. Gene pair : ENSG00000176340 -> ENSG00000184076

Number of common elements: 39

Common elements between set 9 of the local method (MDI) and set 9 of the global method:

1. Gene pair : ENSG00000121281 -> ENSG00000166164

2. Gene pair : ENSG00000081237 -> ENSG00000115232

3. Gene pair : ENSG00000142444 -> ENSG00000161179

4. Gene pair : ENSG00000196126 -> ENSG00000179583

5. Gene pair : ENSG00000169203 -> ENSG00000183604

6. Gene pair : ENSG00000241945 -> ENSG00000160221

7. Gene pair : ENSG00000115232 -> ENSG00000081237

8. Gene pair : ENSG00000161179 -> ENSG00000142444

9. Gene pair : ENSG00000149485 -> ENSG00000134824

10. Gene pair : ENSG00000172057 -> ENSG00000073605

11. Gene pair : ENSG00000160221 -> ENSG00000241945

12. Gene pair : ENSG00000198355 -> ENSG00000152518

13. Gene pair : ENSG00000134824 -> ENSG00000149485

14. Gene pair : ENSG00000073605 -> ENSG00000172057

15. Gene pair : ENSG00000152518 -> ENSG00000151414

16. Gene pair : ENSG00000166164 -> ENSG00000121281

17. Gene pair : ENSG00000172057 -> ENSG00000161395

18. Gene pair : ENSG00000152518 -> ENSG00000198355

19. Gene pair : ENSG00000172543 -> ENSG00000172500

20. Gene pair : ENSG00000183604 -> ENSG00000169203

21. Gene pair : ENSG00000151414 -> ENSG00000152518

Number of common elements: 21

Common elements between set 10 of the local method (MDI) and set 10 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340

2. Gene pair : ENSG00000142444 -> ENSG00000090238

3. Gene pair : ENSG00000142444 -> ENSG00000136240

4. Gene pair : ENSG00000090238 -> ENSG00000142444

5. Gene pair : ENSG00000241945 -> ENSG00000160221

6. Gene pair : ENSG00000268575 -> ENSG00000215790

7. Gene pair : ENSG00000009790 -> ENSG00000102901

8. Gene pair : ENSG00000090238 -> ENSG00000136240

9. Gene pair : ENSG00000149485 -> ENSG00000134824

10. Gene pair : ENSG00000172057 -> ENSG00000073605

11. Gene pair : ENSG00000130592 -> ENSG00000102879

12. Gene pair : ENSG00000136240 -> ENSG00000142444

13. Gene pair : ENSG00000160221 -> ENSG00000241945

14. Gene pair : ENSG00000134824 -> ENSG00000149485

15. Gene pair : ENSG00000073605 -> ENSG00000172057

16. Gene pair : ENSG00000205534 -> ENSG00000169203

17. Gene pair : ENSG00000172500 -> ENSG00000172543

18. Gene pair : ENSG00000176986 -> ENSG00000141367

19. Gene pair : ENSG00000152518 -> ENSG00000198355

20. Gene pair : ENSG00000172543 -> ENSG00000172500

21. Gene pair : ENSG00000215790 -> ENSG00000268575

22. Gene pair : ENSG00000102879 -> ENSG00000130592

23. Gene pair : ENSG00000176340 -> ENSG00000184076

Number of common elements: 23

Common elements between set 11 of the local method (MDI) and set 11 of the global method:

1. Gene pair : ENSG00000196502 -> ENSG00000178952

2. Gene pair : ENSG00000178952 -> ENSG00000184110

3. Gene pair : ENSG00000184076 -> ENSG00000176340

4. Gene pair : ENSG00000241945 -> ENSG00000160221

5. Gene pair : ENSG00000189339 -> ENSG00000215790

6. Gene pair : ENSG00000160221 -> ENSG00000241945

7. Gene pair : ENSG00000134824 -> ENSG00000149485

8. Gene pair : ENSG00000176340 -> ENSG00000184076

9. Gene pair : ENSG00000248333 -> ENSG00000189339

10. Gene pair : ENSG00000215790 -> ENSG00000189339

11. Gene pair : ENSG00000149485 -> ENSG00000134824

12. Gene pair : ENSG00000184110 -> ENSG00000178952

Number of common elements: 12

Common elements between set 12 of the local method (MDI) and set 12 of the global method:

1. Gene pair : ENSG00000168286 -> ENSG00000152518

2. Gene pair : ENSG00000142444 -> ENSG00000090238

3. Gene pair : ENSG00000198355 -> ENSG00000151414

4. Gene pair : ENSG00000142444 -> ENSG00000136240

5. Gene pair : ENSG00000090238 -> ENSG00000142444
6. Gene pair : ENSG00000169203 -> ENSG00000183604
7. Gene pair : ENSG00000102879 -> ENSG00000164054
8. Gene pair : ENSG00000152518 -> ENSG00000168286
9. Gene pair : ENSG00000241945 -> ENSG00000160221
10. Gene pair : ENSG00000102879 -> ENSG00000178952
11. Gene pair : ENSG00000090238 -> ENSG00000136240
12. Gene pair : ENSG00000149485 -> ENSG00000134824
13. Gene pair : ENSG00000168286 -> ENSG00000198355
14. Gene pair : ENSG00000198355 -> ENSG00000168286
15. Gene pair : ENSG00000136240 -> ENSG00000142444
16. Gene pair : ENSG00000160221 -> ENSG00000241945
17. Gene pair : ENSG00000172543 -> ENSG00000122223
18. Gene pair : ENSG00000198355 -> ENSG00000152518
19. Gene pair : ENSG00000134824 -> ENSG00000149485
20. Gene pair : ENSG00000215790 -> ENSG00000189339
21. Gene pair : ENSG00000164054 -> ENSG00000206503
22. Gene pair : ENSG00000102879 -> ENSG00000010256
23. Gene pair : ENSG00000122223 -> ENSG00000172543
24. Gene pair : ENSG00000152518 -> ENSG00000151414
25. Gene pair : ENSG00000152518 -> ENSG00000198355
26. Gene pair : ENSG00000164054 -> ENSG00000102879
27. Gene pair : ENSG00000136240 -> ENSG00000090238
28. Gene pair : ENSG00000183604 -> ENSG00000169203

29. Gene pair : ENSG00000151414 -> ENSG00000152518
30. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 30

Common elements between set 13 of the local method (MDI) and set 13 of the global method:

1. Gene pair : ENSG00000168286 -> ENSG00000152518
2. Gene pair : ENSG00000142444 -> ENSG00000136240
3. Gene pair : ENSG00000196126 -> ENSG00000179583
4. Gene pair : ENSG00000169203 -> ENSG00000183604
5. Gene pair : ENSG00000152518 -> ENSG00000168286
6. Gene pair : ENSG00000241945 -> ENSG00000160221
7. Gene pair : ENSG00000149485 -> ENSG00000134824
8. Gene pair : ENSG00000168286 -> ENSG00000198355
9. Gene pair : ENSG00000172057 -> ENSG00000073605
10. Gene pair : ENSG00000198355 -> ENSG00000168286
11. Gene pair : ENSG00000136240 -> ENSG00000142444
12. Gene pair : ENSG00000160221 -> ENSG00000241945
13. Gene pair : ENSG00000198355 -> ENSG00000152518
14. Gene pair : ENSG00000134824 -> ENSG00000149485
15. Gene pair : ENSG00000073605 -> ENSG00000172057
16. Gene pair : ENSG00000172500 -> ENSG00000172543
17. Gene pair : ENSG00000152518 -> ENSG00000151414
18. Gene pair : ENSG00000152518 -> ENSG00000198355
19. Gene pair : ENSG00000179583 -> ENSG00000196126
20. Gene pair : ENSG00000172543 -> ENSG00000172500

21. Gene pair : ENSG00000215790 -> ENSG00000268575

22. Gene pair : ENSG00000183604 -> ENSG00000169203

23. Gene pair : ENSG00000151414 -> ENSG00000152518

Number of common elements: 23

Common elements between set 14 of the local method (MDI) and set 14 of the global method:

1. Gene pair : ENSG00000172057 -> ENSG00000073605

2. Gene pair : ENSG00000152518 -> ENSG00000151414

3. Gene pair : ENSG00000184076 -> ENSG00000176340

4. Gene pair : ENSG00000151414 -> ENSG00000152518

5. Gene pair : ENSG00000241945 -> ENSG00000160221

6. Gene pair : ENSG00000169567 -> ENSG00000233276

7. Gene pair : ENSG00000179583 -> ENSG00000196126

8. Gene pair : ENSG00000268575 -> ENSG00000215790

9. Gene pair : ENSG00000149485 -> ENSG00000134824

10. Gene pair : ENSG00000160221 -> ENSG00000241945

11. Gene pair : ENSG00000215790 -> ENSG00000268575

12. Gene pair : ENSG00000196126 -> ENSG00000179583

13. Gene pair : ENSG00000134824 -> ENSG00000149485

14. Gene pair : ENSG00000176340 -> ENSG00000184076

15. Gene pair : ENSG00000073605 -> ENSG00000172057

16. Gene pair : ENSG00000215790 -> ENSG00000189339

17. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 17

Common elements between set 15 of the local method (MDI) and set 15 of the global method:

1. Gene pair : ENSG00000205534 -> ENSG00000183604
2. Gene pair : ENSG00000121281 -> ENSG00000166164
3. Gene pair : ENSG00000169203 -> ENSG00000183604
4. Gene pair : ENSG00000081154 -> ENSG00000117500
5. Gene pair : ENSG00000241945 -> ENSG00000160221
6. Gene pair : ENSG00000149485 -> ENSG00000134824
7. Gene pair : ENSG00000172057 -> ENSG00000073605
8. Gene pair : ENSG00000183604 -> ENSG00000205534
9. Gene pair : ENSG00000160221 -> ENSG00000241945
10. Gene pair : ENSG00000172543 -> ENSG00000122223
11. Gene pair : ENSG00000115484 -> ENSG00000144381
12. Gene pair : ENSG00000134824 -> ENSG00000149485
13. Gene pair : ENSG00000073605 -> ENSG00000172057
14. Gene pair : ENSG00000215790 -> ENSG00000189339
15. Gene pair : ENSG00000122223 -> ENSG00000172543
16. Gene pair : ENSG00000166164 -> ENSG00000121281
17. Gene pair : ENSG00000144381 -> ENSG00000115484
18. Gene pair : ENSG00000183604 -> ENSG00000169203
19. Gene pair : ENSG00000150753 -> ENSG00000144381
20. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 20

Common elements between set 16 of the local method (MDI) and set 16 of the global method:

1. Gene pair : ENSG00000178952 -> ENSG00000184110
2. Gene pair : ENSG00000241945 -> ENSG00000160221
3. Gene pair : ENSG00000102901 -> ENSG00000157873
4. Gene pair : ENSG00000136240 -> ENSG00000142444
5. Gene pair : ENSG00000144381 -> ENSG00000115484
6. Gene pair : ENSG00000160221 -> ENSG00000241945
7. Gene pair : ENSG00000183604 -> ENSG00000169203
8. Gene pair : ENSG00000142444 -> ENSG00000136240
9. Gene pair : ENSG00000115484 -> ENSG00000144381
10. Gene pair : ENSG00000134824 -> ENSG00000149485
11. Gene pair : ENSG00000169203 -> ENSG00000183604
12. Gene pair : ENSG00000149485 -> ENSG00000134824
13. Gene pair : ENSG00000184110 -> ENSG00000178952

Number of common elements: 13

Common elements between set 17 of the local method (MDI) and set 17 of the global method:

1. Gene pair : ENSG00000168286 -> ENSG00000152518
2. Gene pair : ENSG00000142444 -> ENSG00000161179
3. Gene pair : ENSG00000142444 -> ENSG00000090238
4. Gene pair : ENSG00000142444 -> ENSG00000136240
5. Gene pair : ENSG00000090238 -> ENSG00000142444
6. Gene pair : ENSG00000169203 -> ENSG00000183604

7. Gene pair : ENSG00000152518 -> ENSG00000168286
8. Gene pair : ENSG00000241945 -> ENSG00000160221
9. Gene pair : ENSG00000161179 -> ENSG00000142444
10. Gene pair : ENSG00000009790 -> ENSG00000102901
11. Gene pair : ENSG00000149485 -> ENSG00000134824
12. Gene pair : ENSG00000168286 -> ENSG00000198355
13. Gene pair : ENSG00000198355 -> ENSG00000168286
14. Gene pair : ENSG00000136240 -> ENSG00000142444
15. Gene pair : ENSG00000160221 -> ENSG00000241945
16. Gene pair : ENSG00000198355 -> ENSG00000152518
17. Gene pair : ENSG00000134824 -> ENSG00000149485
18. Gene pair : ENSG00000102901 -> ENSG00000009790
19. Gene pair : ENSG00000152518 -> ENSG00000151414
20. Gene pair : ENSG00000151151 -> ENSG00000096968
21. Gene pair : ENSG00000152518 -> ENSG00000198355
22. Gene pair : ENSG00000096968 -> ENSG00000151151
23. Gene pair : ENSG00000183604 -> ENSG00000169203
24. Gene pair : ENSG00000151414 -> ENSG00000152518

Number of common elements: 24

Common elements between set 18 of the local method (MDI) and set 18 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000168286 -> ENSG00000152518
3. Gene pair : ENSG00000142444 -> ENSG00000161179
4. Gene pair : ENSG00000261338 -> ENSG00000135926

5. Gene pair : ENSG00000152518 -> ENSG00000168286
6. Gene pair : ENSG00000241945 -> ENSG00000160221
7. Gene pair : ENSG00000161179 -> ENSG00000142444
8. Gene pair : ENSG00000149485 -> ENSG00000134824
9. Gene pair : ENSG00000168286 -> ENSG00000198355
10. Gene pair : ENSG00000172057 -> ENSG00000073605
11. Gene pair : ENSG00000198355 -> ENSG00000168286
12. Gene pair : ENSG00000160221 -> ENSG00000241945
13. Gene pair : ENSG00000010256 -> ENSG00000102879
14. Gene pair : ENSG00000198355 -> ENSG00000152518
15. Gene pair : ENSG00000134824 -> ENSG00000149485
16. Gene pair : ENSG00000073605 -> ENSG00000172057
17. Gene pair : ENSG00000102879 -> ENSG00000010256
18. Gene pair : ENSG00000172500 -> ENSG00000172543
19. Gene pair : ENSG00000166164 -> ENSG00000121281
20. Gene pair : ENSG00000135926 -> ENSG00000261338
21. Gene pair : ENSG00000152518 -> ENSG00000198355
22. Gene pair : ENSG00000172543 -> ENSG00000172500
23. Gene pair : ENSG00000176340 -> ENSG00000184076

Number of common elements: 23

Common elements between set 19 of the local method (MDI) and set 19 of the global method:

1. Gene pair : ENSG00000142444 -> ENSG00000090238
2. Gene pair : ENSG00000090238 -> ENSG00000142444
3. Gene pair : ENSG00000176340 -> ENSG00000184076

4. Gene pair : ENSG00000241945 -> ENSG00000160221
5. Gene pair : ENSG00000268575 -> ENSG00000215790
6. Gene pair : ENSG00000149485 -> ENSG00000134824
7. Gene pair : ENSG00000172057 -> ENSG00000073605
8. Gene pair : ENSG00000268575 -> ENSG00000189339
9. Gene pair : ENSG00000189339 -> ENSG00000268575
10. Gene pair : ENSG00000160221 -> ENSG00000241945
11. Gene pair : ENSG00000178952 -> ENSG00000010256
12. Gene pair : ENSG00000134824 -> ENSG00000149485
13. Gene pair : ENSG00000073605 -> ENSG00000172057
14. Gene pair : ENSG00000215790 -> ENSG00000189339
15. Gene pair : ENSG00000122223 -> ENSG00000172543
16. Gene pair : ENSG00000010256 -> ENSG00000178952
17. Gene pair : ENSG00000102901 -> ENSG00000143224
18. Gene pair : ENSG00000215790 -> ENSG00000268575
19. Gene pair : ENSG00000183604 -> ENSG00000169203
20. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 20

Common elements between set 20 of the local method (MDI) and set 20 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000081237 -> ENSG00000115232
3. Gene pair : ENSG00000198355 -> ENSG00000151414
4. Gene pair : ENSG00000176340 -> ENSG00000184076
5. Gene pair : ENSG00000169203 -> ENSG00000183604

6. Gene pair : ENSG00000241945 -> ENSG00000160221
7. Gene pair : ENSG00000115232 -> ENSG00000081237
8. Gene pair : ENSG00000268575 -> ENSG00000215790
9. Gene pair : ENSG00000149485 -> ENSG00000134824
10. Gene pair : ENSG00000172057 -> ENSG00000073605
11. Gene pair : ENSG00000160221 -> ENSG00000241945
12. Gene pair : ENSG00000172543 -> ENSG00000122223
13. Gene pair : ENSG00000134824 -> ENSG00000149485
14. Gene pair : ENSG00000248333 -> ENSG00000189339
15. Gene pair : ENSG00000171700 -> ENSG00000102879
16. Gene pair : ENSG00000073605 -> ENSG00000172057
17. Gene pair : ENSG00000215790 -> ENSG00000189339
18. Gene pair : ENSG00000122223 -> ENSG00000172543
19. Gene pair : ENSG00000172057 -> ENSG00000161395
20. Gene pair : ENSG00000102879 -> ENSG00000171700
21. Gene pair : ENSG00000215790 -> ENSG00000268575
22. Gene pair : ENSG00000102879 -> ENSG00000130592
23. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 23

Common elements between set 21 of the local method (MDI) and set 21 of the global method:

1. Gene pair : ENSG00000114395 -> ENSG00000102879
2. Gene pair : ENSG00000241945 -> ENSG00000160221
3. Gene pair : ENSG00000149485 -> ENSG00000134824
4. Gene pair : ENSG00000168286 -> ENSG00000198355

5. Gene pair : ENSG00000196502 -> ENSG00000178952
6. Gene pair : ENSG00000172057 -> ENSG00000073605
7. Gene pair : ENSG00000198355 -> ENSG00000168286
8. Gene pair : ENSG00000189339 -> ENSG00000268575
9. Gene pair : ENSG00000160221 -> ENSG00000241945
10. Gene pair : ENSG00000134824 -> ENSG00000149485
11. Gene pair : ENSG00000073605 -> ENSG00000172057
12. Gene pair : ENSG00000172500 -> ENSG00000172543
13. Gene pair : ENSG00000152518 -> ENSG00000198355
14. Gene pair : ENSG00000164054 -> ENSG00000102879
15. Gene pair : ENSG00000178952 -> ENSG00000196502
16. Gene pair : ENSG00000172543 -> ENSG00000172500
17. Gene pair : ENSG00000183604 -> ENSG00000169203
18. Gene pair : ENSG00000151414 -> ENSG00000152518
19. Gene pair : ENSG00000176340 -> ENSG00000184076

Number of common elements: 19

Common elements between set 22 of the local method (MDI) and set 22 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000081237 -> ENSG00000115232
3. Gene pair : ENSG00000068831 -> ENSG00000149923
4. Gene pair : ENSG00000176340 -> ENSG00000184076
5. Gene pair : ENSG00000169203 -> ENSG00000183604
6. Gene pair : ENSG00000144381 -> ENSG00000150753
7. Gene pair : ENSG00000241945 -> ENSG00000160221

8. Gene pair : ENSG00000115232 -> ENSG00000081237
9. Gene pair : ENSG00000115484 -> ENSG00000150753
10. Gene pair : ENSG00000196502 -> ENSG00000178952
11. Gene pair : ENSG00000136240 -> ENSG00000142444
12. Gene pair : ENSG00000160221 -> ENSG00000241945
13. Gene pair : ENSG00000010256 -> ENSG00000102879
14. Gene pair : ENSG00000115484 -> ENSG00000144381
15. Gene pair : ENSG00000134824 -> ENSG00000149485
16. Gene pair : ENSG00000171700 -> ENSG00000102879
17. Gene pair : ENSG00000215790 -> ENSG00000189339
18. Gene pair : ENSG00000150753 -> ENSG00000115484
19. Gene pair : ENSG00000102879 -> ENSG00000010256
20. Gene pair : ENSG00000102879 -> ENSG00000171700
21. Gene pair : ENSG00000178952 -> ENSG00000196502
22. Gene pair : ENSG00000144381 -> ENSG00000115484
23. Gene pair : ENSG00000183604 -> ENSG00000169203
24. Gene pair : ENSG00000150753 -> ENSG00000144381
25. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 25

Common elements between set 23 of the local method (MDI) and set 23 of the global method:

1. Gene pair : ENSG00000184076 -> ENSG00000176340
2. Gene pair : ENSG00000205534 -> ENSG00000183604
3. Gene pair : ENSG00000168286 -> ENSG00000152518
4. Gene pair : ENSG00000142444 -> ENSG00000136240

5. Gene pair : ENSG00000176340 -> ENSG00000184076
6. Gene pair : ENSG00000141367 -> ENSG00000176986
7. Gene pair : ENSG00000152518 -> ENSG00000168286
8. Gene pair : ENSG00000241945 -> ENSG00000160221
9. Gene pair : ENSG00000149485 -> ENSG00000134824
10. Gene pair : ENSG00000172046 -> ENSG00000149923
11. Gene pair : ENSG00000168286 -> ENSG00000198355
12. Gene pair : ENSG00000198355 -> ENSG00000168286
13. Gene pair : ENSG00000183604 -> ENSG00000205534
14. Gene pair : ENSG00000149923 -> ENSG00000172046
15. Gene pair : ENSG00000136240 -> ENSG00000142444
16. Gene pair : ENSG00000160221 -> ENSG00000241945
17. Gene pair : ENSG00000010256 -> ENSG00000102879
18. Gene pair : ENSG00000198355 -> ENSG00000152518
19. Gene pair : ENSG00000134824 -> ENSG00000149485
20. Gene pair : ENSG00000215790 -> ENSG00000189339
21. Gene pair : ENSG00000102879 -> ENSG00000010256
22. Gene pair : ENSG00000172500 -> ENSG00000172543
23. Gene pair : ENSG00000176986 -> ENSG00000141367
24. Gene pair : ENSG00000152518 -> ENSG00000198355
25. Gene pair : ENSG00000164054 -> ENSG00000102879
26. Gene pair : ENSG00000172543 -> ENSG00000172500
27. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 27

Common elements between set 24 of the local method (MDI) and set 24 of the global method:

1. Gene pair : ENSG00000196502 -> ENSG00000178952
2. Gene pair : ENSG00000189339 -> ENSG00000248333
3. Gene pair : ENSG00000152518 -> ENSG00000198355
4. Gene pair : ENSG00000241945 -> ENSG00000160221
5. Gene pair : ENSG00000115232 -> ENSG00000081237
6. Gene pair : ENSG00000178952 -> ENSG00000196502
7. Gene pair : ENSG00000149923 -> ENSG00000172046
8. Gene pair : ENSG00000189339 -> ENSG00000215790
9. Gene pair : ENSG00000081237 -> ENSG00000115232
10. Gene pair : ENSG00000149485 -> ENSG00000134824
11. Gene pair : ENSG00000136240 -> ENSG00000142444
12. Gene pair : ENSG00000160221 -> ENSG00000241945
13. Gene pair : ENSG00000198355 -> ENSG00000151414
14. Gene pair : ENSG00000215790 -> ENSG00000268575
15. Gene pair : ENSG00000134824 -> ENSG00000149485
16. Gene pair : ENSG00000248333 -> ENSG00000189339
17. Gene pair : ENSG00000215790 -> ENSG00000189339
18. Gene pair : ENSG00000172046 -> ENSG00000149923

Number of common elements: 18

Common elements between set 25 of the local method (MDI) and set 25 of the global method:

1. Gene pair : ENSG00000151151 -> ENSG00000096968

2. Gene pair : ENSG00000184076 -> ENSG00000176340
3. Gene pair : ENSG00000241945 -> ENSG00000160221
4. Gene pair : ENSG00000096968 -> ENSG00000151151
5. Gene pair : ENSG00000136240 -> ENSG00000142444
6. Gene pair : ENSG00000149485 -> ENSG00000134824
7. Gene pair : ENSG00000160221 -> ENSG00000241945
8. Gene pair : ENSG00000268575 -> ENSG00000215790
9. Gene pair : ENSG00000183604 -> ENSG00000169203
10. Gene pair : ENSG00000142444 -> ENSG00000136240
11. Gene pair : ENSG00000144381 -> ENSG00000115484
12. Gene pair : ENSG00000134824 -> ENSG00000149485
13. Gene pair : ENSG00000215790 -> ENSG00000268575
14. Gene pair : ENSG00000169203 -> ENSG00000183604
15. Gene pair : ENSG00000176340 -> ENSG00000184076
16. Gene pair : ENSG00000215790 -> ENSG00000189339
17. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 17

Common elements between set 26 of the local method (MDI) and set 26 of the global method:

1. Gene pair : ENSG00000178952 -> ENSG00000184110
2. Gene pair : ENSG00000081237 -> ENSG00000115232
3. Gene pair : ENSG00000169203 -> ENSG00000183604
4. Gene pair : ENSG00000144381 -> ENSG00000150753
5. Gene pair : ENSG00000241945 -> ENSG00000160221
6. Gene pair : ENSG00000115232 -> ENSG00000081237

7. Gene pair : ENSG00000115484 -> ENSG00000150753
8. Gene pair : ENSG00000149485 -> ENSG00000134824
9. Gene pair : ENSG00000102901 -> ENSG00000157873
10. Gene pair : ENSG00000136240 -> ENSG00000142444
11. Gene pair : ENSG00000160221 -> ENSG00000241945
12. Gene pair : ENSG00000115484 -> ENSG00000144381
13. Gene pair : ENSG00000134824 -> ENSG00000149485
14. Gene pair : ENSG00000215790 -> ENSG00000189339
15. Gene pair : ENSG00000184110 -> ENSG00000178952
16. Gene pair : ENSG00000144381 -> ENSG00000115484
17. Gene pair : ENSG00000183604 -> ENSG00000169203
18. Gene pair : ENSG00000150753 -> ENSG00000144381
19. Gene pair : ENSG00000189339 -> ENSG00000215790

Number of common elements: 19