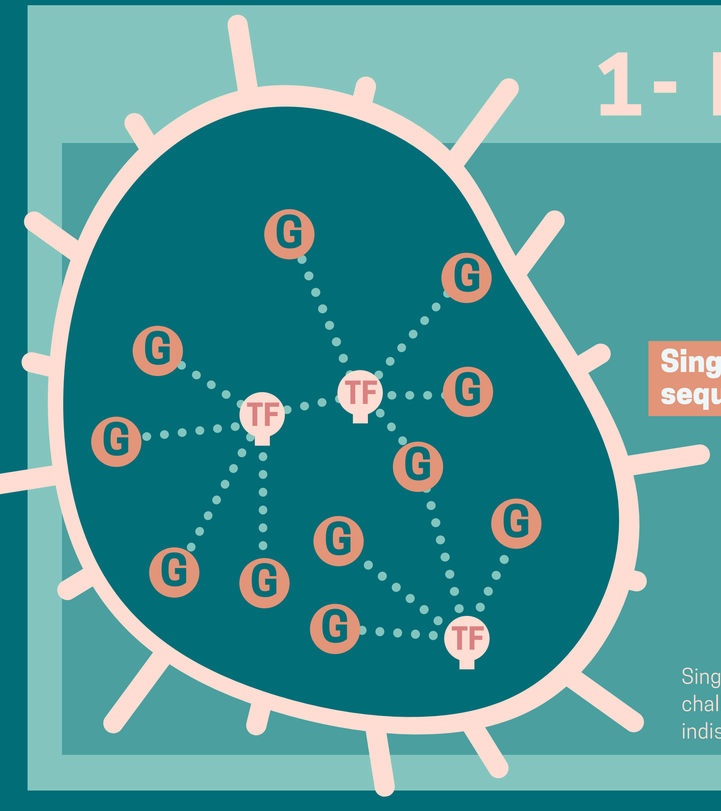
Semi-supervised learning for tree-based regressors to improve the prediction of the interactions between genes

Lisa Chabrier, Anton Crombach, Sergio Peignier, Christophe Rigotti

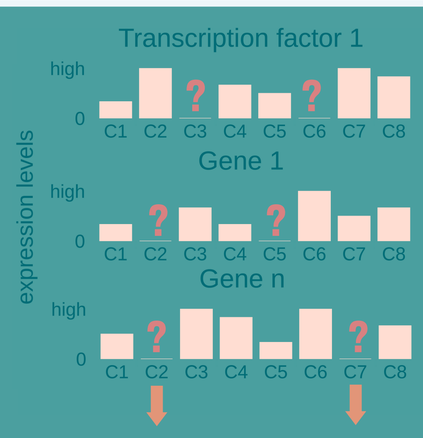
Semi-supervised learning for tree-based regressors to improve the prediction of the interactions between genes

Lisa Chabrier[1], Anton Crombach[1], Sergio Peignier[2], Christophe Rigotti[1]
[1]: Univ Lyon, Inria, INSA Lyon, CNRS, UCBL, LIRIS (UMR5205), [2]: Univ Lyon, INSA Lyon, INRAE, BF2I (UMR203)

1- INFERRING GENE REGULATORY NETWORKS

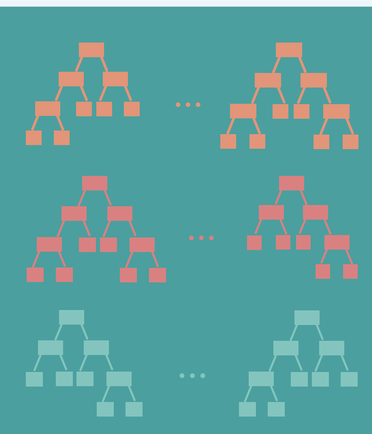


Single cell RNA sequencing

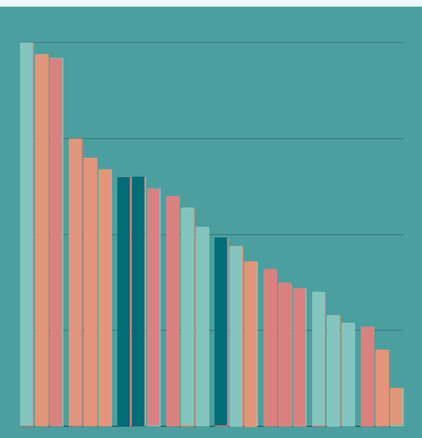


Single-cell RNA-sequencing measures the expression of genes cell by cell. A major challenge in the analysis of these data is so-called dropouts, that are indistinguishable from real zeros. And zeros can represent up to 95% of the data.

Arboreto framework [1]



Regression models built from single-cell data, predicting the expression of one target gene from the expression of transcription factors (TFs).

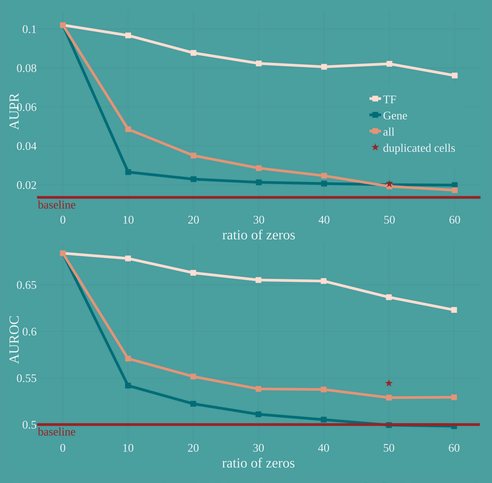


Feature ranking is computed for each model and used to score the interaction between the TF (the feature) and target gene.

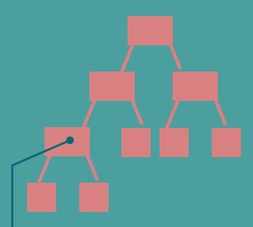
QUESTIONS

- **How to build tree-based models that integrate measured values and handle the uncertain origins of zeros?**
- **How to efficiently build the tree regression models in order to use the algorithm in the Arboreto pipeline?**
- **Can we find a unique optimal value of the parameter for all single-cell datasets?**

2- SEMI-SUPERVISED LEARNING FOR TREES



Experiment on the DREAM5 dataset [2]. For each curve, dropouts were added to a different part of the data: TFs, non-TF genes, and the whole dataset.



Splits are composed of two elements: the feature and the threshold with max reduction of the impurity in child nodes. In scikit-learn [3], the Python library used in Arboreto [1], the impurity of a node is based on the mean squared error.

Criterions

MSE

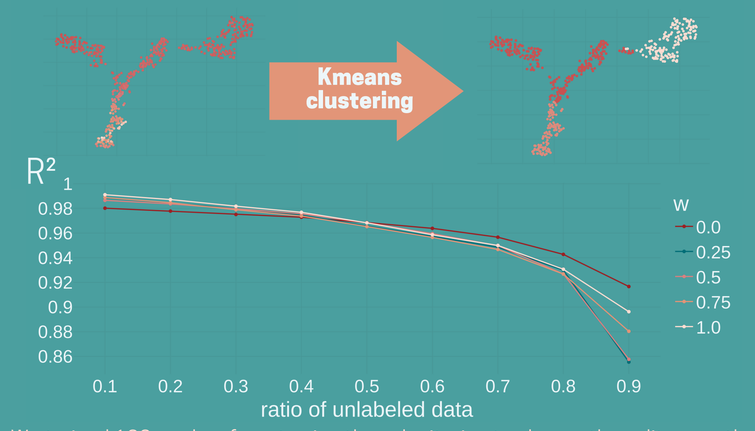
$$imp_{mse} = Var^{Y}(E_l)$$

	TF1	TF2	G1	T	Gn	
cell 3	0.00	0.00	3.61	4.34	1.25	Labeled Examples
cell 4	0.00	0.00	3.47	4.91	3.71	
cell 5	0.00	0.00	2.34	1.97	3.76	
cell 8	2.60	4.34	6.07	4.81	0.00	
cell 1	3.43	0.00	0.00	0.00	1.61	Unlabeled Examples
cell 2	2.85	3.79	0.00	0.00	2.11	
cell 6	0.00	0.00	0.00	0.00	0.00	
cell 7	2.92	0.00	0.00	0.00	1.52	

SSL

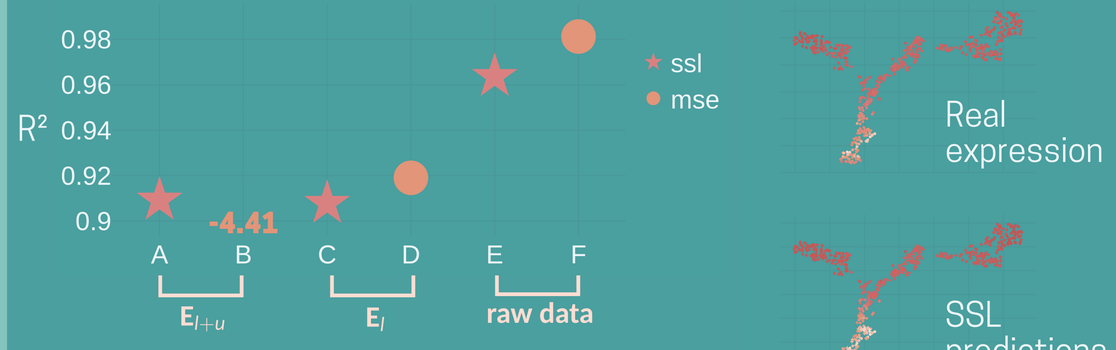
$$imp_{ssl} = w * \frac{Var^{Y}(E_l)}{Var^{Y}(E_l^{train})} + \frac{1-w}{D} * \sum_{i=1}^{D} \frac{Var^{X_i}(E_{l+u})}{Var^{X_i}(E_{l+u}^{train})}$$

Levatic et al. [4] proposed to implement semi-supervised learning for regression trees by replacing the impurity computation with one that takes into account unlabeled examples.



We trained 100 random forest using the ssl criterion to choose the splits over the dataset, with different ratio of unlabeled values and different values for the parameter w. As expected, the R^2 decreases when the number of labeled examples increases, but performance remains extremely high across a wide range of % unlabeled data.

3- RESULTS



For this experiment, we choose one feature to be the target function, and we trained 100 regression forest for each experimental setting decribed by the couple (Impurity measurement, training data). On the figure is represented the mean R^2 score for each experimental setting

4- LIMITS

COMPUTATION TIME

- Scikit-learn implements a so-called impurity proxy that has the same monotony as the real impurity. This reduces the time needed to find the optimal split at each node. For now, our implementation in scikit-learn [3] does not come with such a proxy, but finding and implementing one could reduce the computational power needed.
- Reducing the number of features on which to compute the second term is also a possibility by using a random subset of features.

NEW PARAMETER

In this impurity definition, a new parameter is introduced: w. Tuning this parameter might be an issue, but according to the original paper [4] and the early results we have, it is possible that the value of the parameter will not have such a big influence in our experimental setting. This is still an open question.

REAL SETTING

The implementation of this impurity measure still needs to be tested in a realistic situation, to reproduce the experiment presented in Section 2. This is time-consuming but necessary to answer the initial questions.

[1] T. Moerman, S. Aibar Santos, C. Bravo González-Blas, et al., "**GRNBoost2 andArboreto: Efficient and scalable inference of gene regulatory networks**", Bioinformatics, Jun. 2019, issn: 1367-4803,1460-2059, doi:10.1093/bioinformatics/bty916.

[2] D. Marbach, J. C. Costello, R. Küffner, et al., "**Wisdom of crowds for robust gene network inference**", Nature methods, Jul. 2012, issn:1548-7091. doi:10.1038/nmeth.2016.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort,et al., "**Scikit-learn: Machine learningin Python**", Journal of Machine Learning Research, 2011.

[4] J. Levatić, M. Ceci, T. Stepišnik, et al., "**Semi-supervised regression trees with application to QSAR modelling**", Expert Systems with Applications, Nov. 2020, doi: 10.1016/j.eswa.2020.113569.