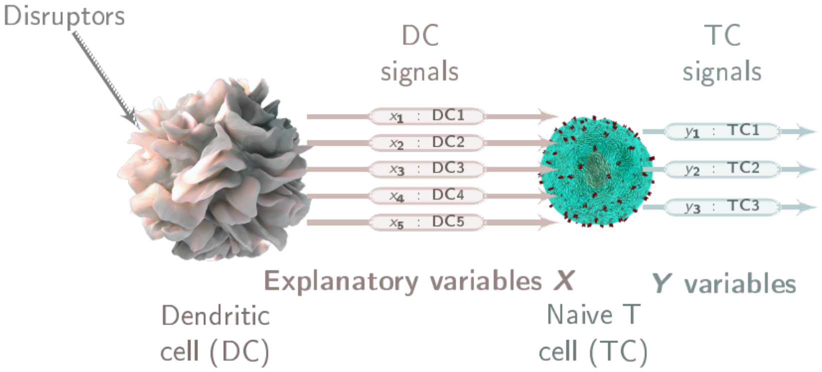
Interpretable Random Forest with Linear Models

Lucie Brolon, Fabien Crauste, Marie Perrot-Dockès

MAP5, Université Paris Cité

Rochebrune 2026

Dialogue between T cells and dendritic cells



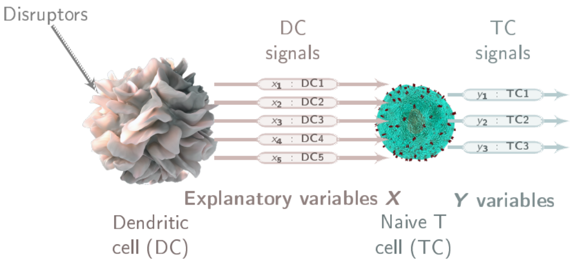
Experiments on human blood

Figure: FIOUUUUUUUUUU

Context — T cells and dendritic cells relationship

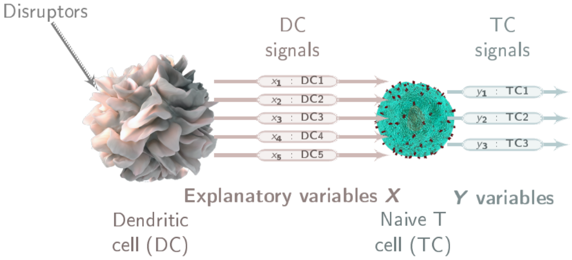
Analyze the *'language'* used by dendritic cells and T cells to communicate with each other:

- types of DC
- types of signals emitted by DC
- simultaneous signals of different natures produced by multiple DC



Project objectives: context specific model, easily interpretable and testable

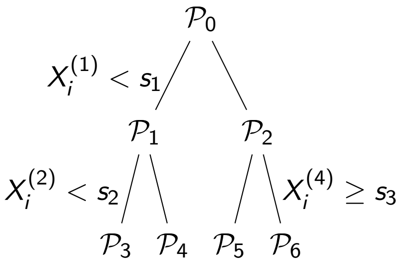
Objectives



- Context specific (relationship between response and explanatory depend on the other explanatory) Tree ?
- Stable rules Forest ?
- Easily interpretable and testable SIRUS?

Context specific : Decision Tree algorithm

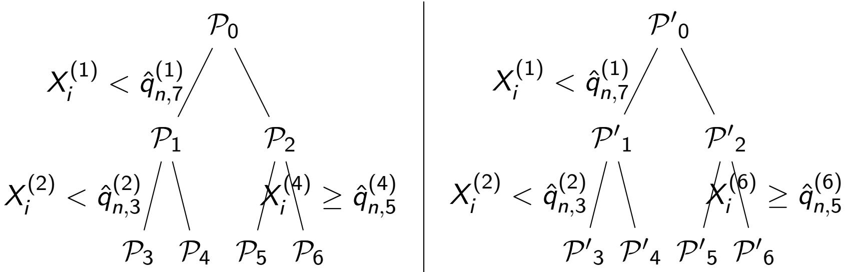
1. Determine the explanatory variables and the threshold that best separate the data.
2. Distribute data accross nodes. Repeat step 1 in each node.
3. Stop the algorithm when the stopping criterion is reached.
4. Estimate $\hat{y}$ by averaging y in each node



SIRUS: Stable and Interpretable RUle Set[1]

Algorithm objective: a random forest to build a set of stable, interpretable decision rules with good predictive capabilities.

- **Step 1: build a forest** (with empirical decile as split values).
- Step 2: select rules according to their appearance frequency in the forest
- Step 3: post-processing : remove rules that are linear combination of more frequent rules (and equal rules).
- Step 4: rules agregation.

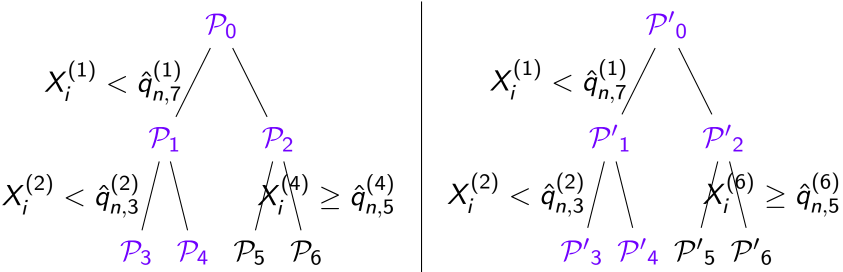


[1] Bénard, Biau, Da Veiga, Scornet (2021)

SIRUS: Stable and Interpretable RUle Set[2]

Algorithm objective: a random forest to build a set of stable, interpretable decision rules with good predictive capabilities.

- Step 1: build a forest (with empirical decile as split values).
- **Step 2: select rules** according to their appearance frequency in the forest
- Step 3: post-processing : remove rules that are linear combination of more frequent rules (and equal rules).
- Step 4: rules agregation.

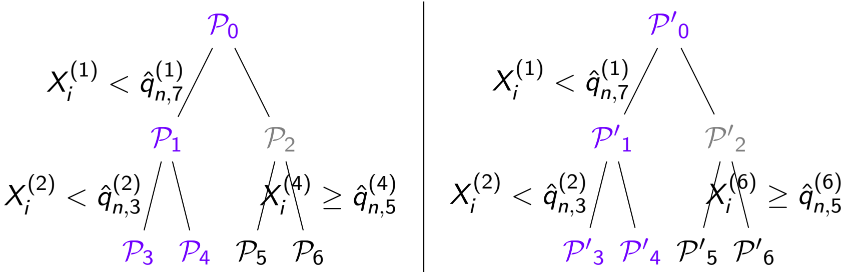


[2] Bénard, Biau, Da Veiga, Scornet (2021)

SIRUS: Stable and Interpretable RUle Set[3]

Algorithm objective: a random forest to build a set of stable, interpretable decision rules with good predictive capabilities.

- Step 1: build a forest (with empirical decile as split values).
- Step 2: select rules according to their appearance frequency in the forest
- **Step 3: post-processing** : remove rules that are linear combination of more frequent rules (and equal rules).
- Step 4: rules agregation.



[3] Bénard, Biau, Da Veiga, Scornet (2021)

SIRUS: Stable and Interpretable RUle Set[4]
Algorithm objective: a random forest to build a set of stable, interpretable decision rules with good predictive capabilities.

- Step 1: build a forest (with empirical decile as split values).
- Step 2: select rules according to their appearance frequency in the forest
- Step 3: post-processing : remove rules that are linear combination of more frequent rules (and equal rules).
- **Step 4: rules agregation.**

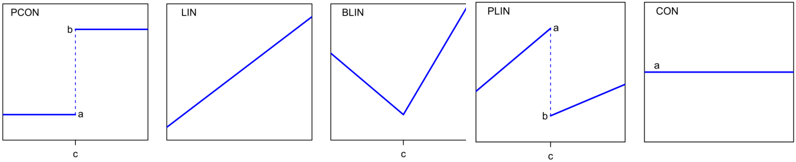
$$(\hat{\beta}_0, \hat{\beta}) = \mathrm{argmin}_{\beta,\beta_0} \| Y - \beta_0 1_n - \beta \Gamma_{n,p_0}, \| + \lambda \|\beta\|_2^2,$$

with Γ_{n,p_0} a $n \times R$ matrix containing the rules values.

[4] Bénard, Biau, Da Veiga, Scornet (2021)

Decision Tree, PILOT algorithm

PILOT[5]: PIecewise Linear Organic Tree, decision tree in which each node can be associated with a different model.



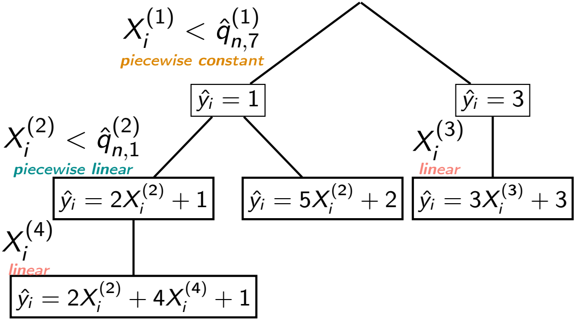
For each node choose the variable $X^{(j)}$ and type t that minimise

$$nlog\left(\frac{RSS(X^{(j)},t)}{n}\right)+\nu log(n),\ \text{with}$$

- $RSS(X^{(j)},t)$ residuals sum of square after the node
- ν a penalty parameters depending on t

[5] Raymaekers, Rousseeuw, Verdonck, Yao (2024)

PILOT[6]: PIecewise Linear Organic Tree, decision tree in which each node can be associated with a different model.



[6] Raymaekers, Rousseeuw, Verdonck, Yao (2024)

Random Forest

Decision tree: simple structure but instability.

Random Forest: set of decision trees.

- each tree is built on a subsample of the data
- for each node, the algorithm selects a subset of the explanatory variables

$\Rightarrow$ Powerful predictivity, less instability.
$\Rightarrow$ BUT *black box*: high number of operations.

Random Forest with PILOT tree: **Ra**ndom **F**orest **F**eaturing **L**inear **E**xtensions (RaFFLE) [7].

[7] Raymaekers, Rousseeuw, Servotte, Verdonck, Yao (2025)

Decision tree: simple structure but instability.

Random Forest: set of decision trees.

- each tree is built on a subsample of the data
- for each node, the algorithm selects a subset of the explanatory variables

⇒ Powerful predictivity, less instability.
⇒ BUT *black box*: high number of operations.

Random Forest with PILOT tree: **Ra**ndom **F**orest **F**eaturing **L**inear **E**xtensions (RaFFLE) [7].

[7] Raymaekers, Rousseeuw, Servotte, Verdonck, Yao (2025)

Stable **P**iecewise **L**inear **I**nterpretable **M**odel

Main idea Extract interpretable rules from a forest of PILOT trees.

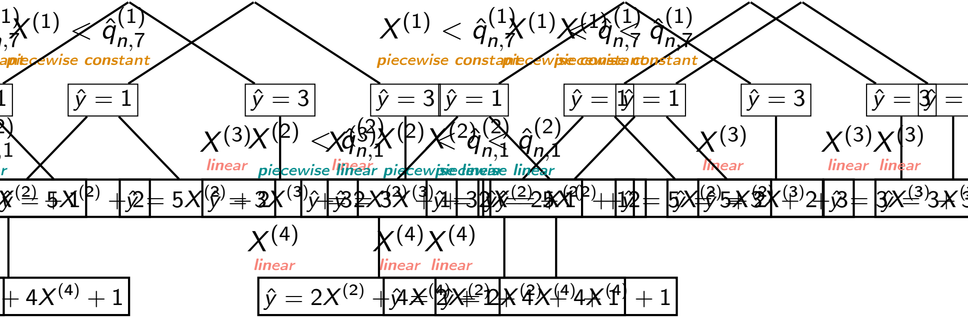
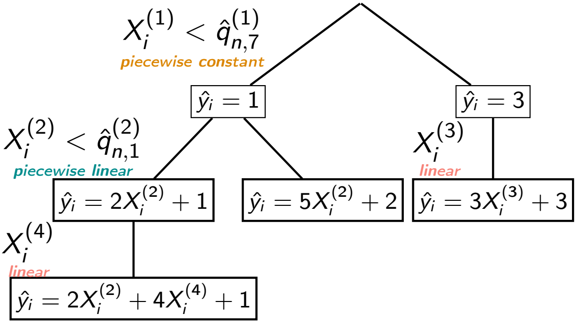


Figure: Glouuups on n'y voit rien !

Difference from SIRUS : rules are not picewise constant.



Rules (Variables)

- $\mathbb{1}_{X^{(1)} < \hat{q}_{n,7}^{(1)}}$
- $\mathbb{1}_{X^{(1)} < \hat{q}_{n,7}^{(1)}} \mathbb{1}_{X^{(2)} < \hat{q}_{n,1}^{(2)}}$
- $\mathbb{1}_{X^{(1)} < \hat{q}_{n,7}^{(1)}} \mathbb{1}_{X^{(2)} < \hat{q}_{n,1}^{(2)}} X^{(2)}$
- $\mathbb{1}_{X^{(1)} < \hat{q}_{n,7}^{(1)}} \mathbb{1}_{X^{(2)} < \hat{q}_{n,1}^{(2)}} X^{(4)}$
- $\mathbb{1}_{X^{(1)} < \hat{q}_{n,7}^{(1)}} \mathbb{1}_{X^{(2)} \geq \hat{q}_{n,1}^{(2)}} X^{(2)}$
- $\mathbb{1}_{X^{(1)} \geq \hat{q}_{n,7}^{(1)}}$
- $\mathbb{1}_{X^{(1)} \geq \hat{q}_{n,7}^{(1)}} X^{(3)}$

Proposed algorithm: SPLIM

Stable **P**iecewise **L**inear **I**nterpretable **M**odel

Algorithm 1 SPLIM

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, number of trees T, frequency threshold s

Output: Aggregated predictor $\hat{m}(\cdot)$

Initialize an empty multiset of rules $\mathcal{R}$
for $t = 1$ to T **do**
    Draw a bootstrap sample $\mathcal{D}_t$ from $\mathcal{D}$
    Fit a PILOT tree on $\mathcal{D}_t$
    Extract all rules from the fitted tree and add them to $\mathcal{R}$
end for
Compute the empirical frequency of each rule $r \in \mathcal{R}$
Retain rules whose empirical frequency exceeds s
Decompose retained rules into sub-rules and fit a ridge regression

EXPERIMENTS

Simulation study — Framework

$$X \in \mathbb{R}^{n,p} \quad , X \sim \mathcal{N}(0, \Sigma)$$

$$Y = f(X) + \epsilon \quad , \epsilon \sim \mathcal{N}(0, \sigma)$$

Parameters:

- n number of observations, p number of explanatory variables
- σ noise on Y, correlation matrix Σ
- data simulation configuration (function f)

Simulation study — Framework

with $f(X)$ corresponding to several configurations:

Scheme name	**Definition of $f(X)$**
CSTE	$\begin{cases} 1, & \text{if } X_1 > q_{0.5,X_1}, \\ 0, & \text{otherwise.} \end{cases}$
REG3	$\begin{cases} X_3 + 1.2, & \text{if } X_1 > q_{0.5,X_1} \text{ and } X_2 > q_{0.5,X_2}, \\ 0.7, & \text{otherwise.} \end{cases}$
REG5	$\begin{cases} X_1 + X_2 + 1.2, & \text{if } X_1 > q_{0.5,X_1}, \\ X_3, & \text{otherwise.} \end{cases}$
EXP	$\begin{cases} \exp(X^{(2)}) + 1.2, & \text{if } X^{(1)} > q_{0.5,X_1}, \\ \exp(X^{(3)}), & \text{otherwise.} \end{cases}$

RMSE: Root Mean Square Error on test sample.
True Positive Rate: proportion of variables correctly included in the model.
False Positive Rate: ... wrongly ...

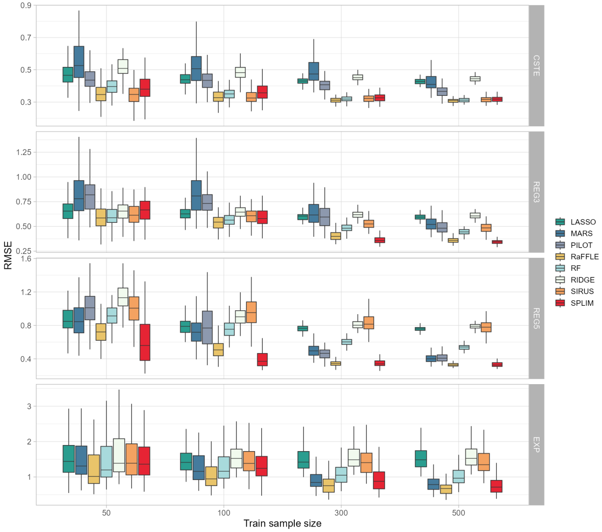


Figure: Root Mean Square Error (RMSE) on test samples across 100 batches, comparing different algorithms.

Simulation study — TPR/FPR comparison

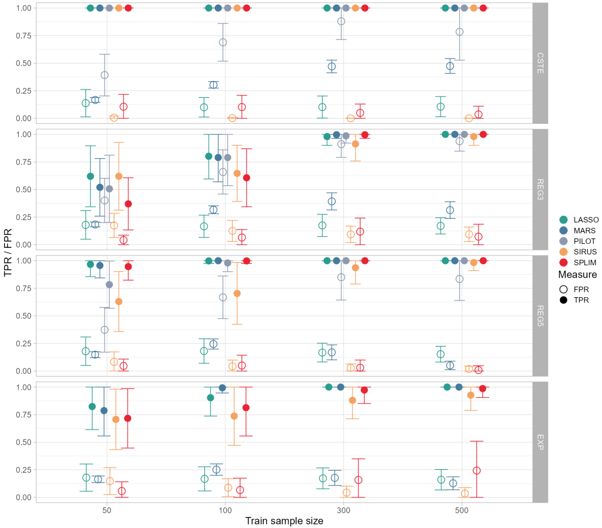


Figure: True and False Positive Rates on test samples across 100 batches, comparing different algorithms.

Stability

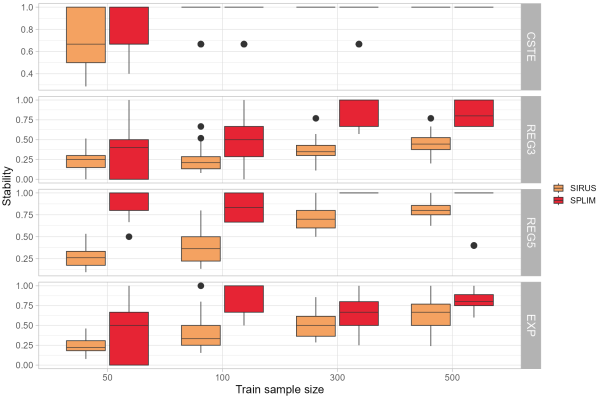


Figure: Stability on 10 fold CV dataset.

$$S(R_1, R_2) = \frac{2 \times |\mathcal{R}_1 \cap \mathcal{R}_2|}{|\mathcal{R}_1| + |\mathcal{R}_2|}$$

Conclusions & Perspectives

Conclusions:

- SPLIM is well-suited to the problem, and perform similarly to or better than SIRUS and RaFFLE.

Perspectives:

- Wait for feedback from the biologists
- Build a multivariate algoritm

Thanks

Thank you for your attention!