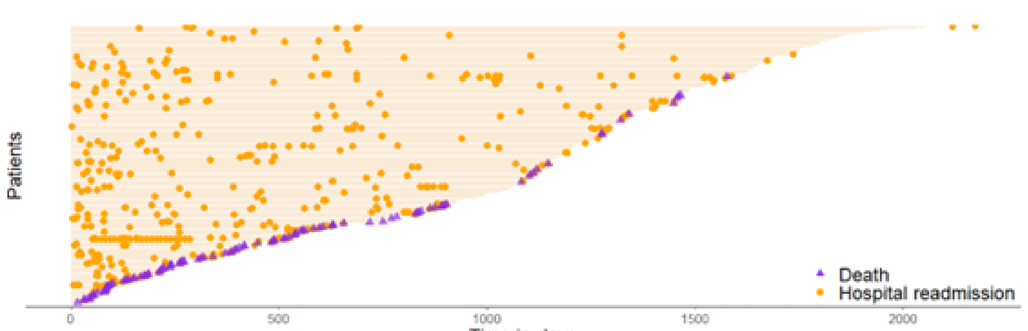
Rencontres R
12 | 14 JUNE

RecForest: Random survival forests to analyze recurrent events in R

Juliette Murris*, Sandrine Katsahian†, Audrey Lavenu‡

** HeKA, Inserm, Inria, Université Paris Cité, Pierre Fabre R&D, †Centre d'Investigation Clinique 1418 Épidémiologie Clinique, Paris, France ‡ Institut de Recherche Mathématique de Rennes (IRMAR), Rennes, France*

MOTIVATING EXAMPLE



Available options within a survival framework

- Time-to-first event (either readmission or death)
- Time-to-reccurence, with or without death

The advent of machine learning

- Usual machine learning algorithms have been extended to account for survival data
- But not to account for survival data and recurrent events, with or without a terminal event.

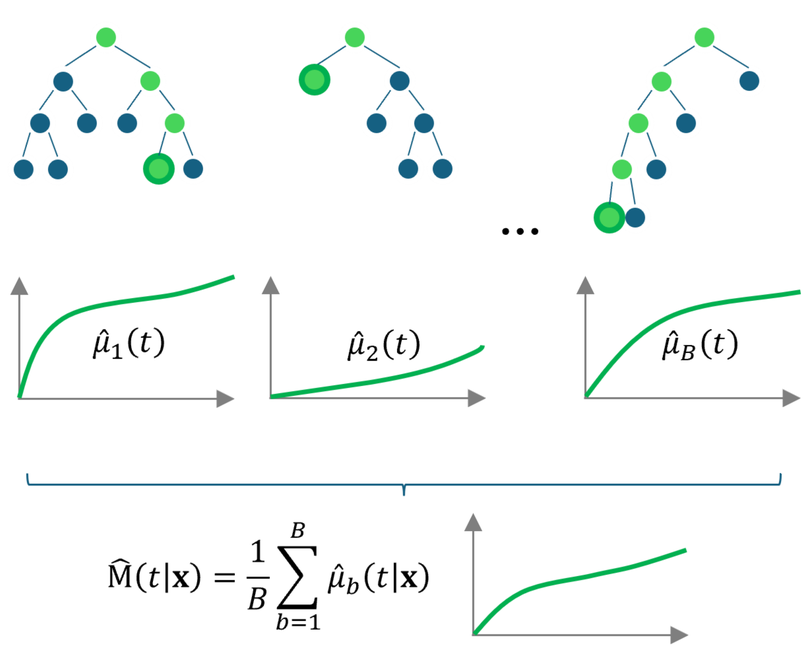
Objectives

- Introduce a **new approach** to model recurrent events using **ensemble methods**
- Introduce **associated R code**

METHODS

RecForest Algorithm

	Without a terminal event	**With a terminal event**
(1) Draw B **bootstrap** samples from the learning data;		
(2) Grow a **survival tree** b extended to recurrent events;		
Splitting rule	Maximize the test statistic	
At each node, $mtry$ predictors are randomly selected with $mtry \in \mathbb{N}$	Pseudo score test from NP estimates	Wald test from Ghosh-Lin model
Terminal node estimator for tree b	$\hat{\mu}_b(t\|\mathbf{x}) = \hat{R}_b(t\|\mathbf{x}) = \int_0^t \frac{N_b(\mathrm{d}u\|\mathbf{x})}{Y_b(\mathrm{d}u\|\mathbf{x})}$	$\hat{\mu}_b(t\|\mathbf{x}) = \int_0^t \hat{S}_b(u\|\mathbf{x})\mathrm{d}\hat{R}_b(u\|\mathbf{x})$
Pruning strategy	A minimal number of events and/or a minimal number of individuals	
(3) **Estimate** $\widehat{\mathrm{M}}$ is computed over the B trees.		



USING R

Step 1 Create a RecForest object

```
mtry <- 5 # number of candidate variables randomly drawn at each node
minsplit <- 5 # minimal number of events required to split the node
nodesize <- 5 # minimal number of subjects required in both child nodes to split
n_trees <- 100 # number of trees
method <- "GL" # Ghosh-Lin for recurrent events, with a terminal event
params <- list(seed = seed, mtry = mtry, minsplit = minsplit, nodesize = nodesize,
    method = method, n_trees = n_trees)
my_recforest <- RecForest(X = X, Y = Y, params = params)
```

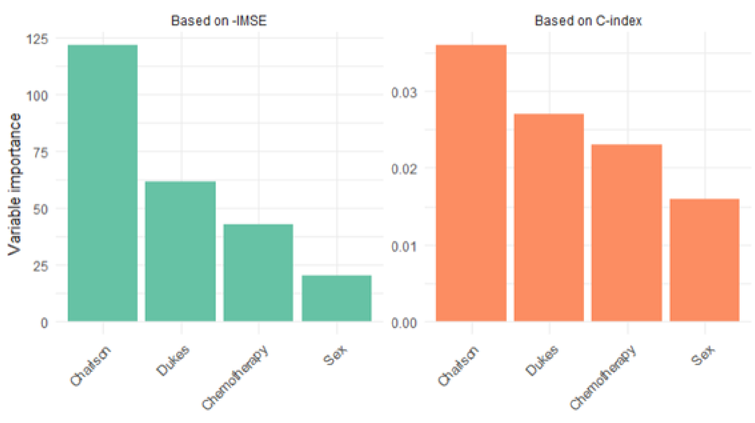
Step 2 Evaluate performances, using adapted versions of C-index and MSE

```
c_index(my_recforest, X_new = NULL) # X_new = NULL refers to OOB samples
mse(my_recforest, X_new = NULL) # X_new = NULL refers to OOB samples
```

Metric	Np	GL1	GL2	GL3	GL4	RecForest	GL*
C-index ↑	0.58 (0.05)	0.53 (0.08)	0.48 (0.08)	0.48 (0.07)	0.45 (0.05)	**0.80 (0.04)**	0.60 (0.06)
IMSE ↓	7 883.50 (6 229.47)	7 843.99 (6 106.36)	8 361.16 (6 292.29)	8 229.08 (6 478.35)	9 981.50 (6 064.23)	**706.02 (508.96)**	7 934.28 (6 606.23)

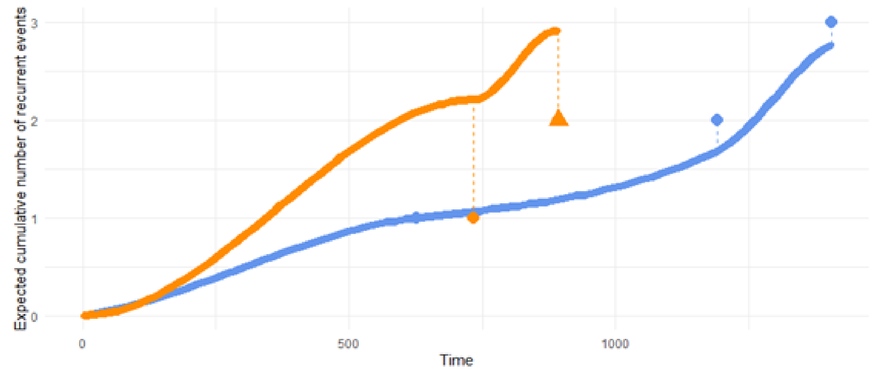
Step 3 Assess variable importance to measure impact on predictions

```
vimp(my_recforest, n_permutations = 10)
```



Step 4 Predict with new data

```
predictions = predict(my_recforest, X_new = X_new)
```



DISCUSSION & CONCLUSION

- Our approach is **simple** and easily **accessible**
- Solid baseline for many **extensions**

Perspectives

- Develop an **R package**

RecForest is a **valuable contribution** for analysing **recurrent events** in **medical** research

BIBLIOGRAPHY

Andrews DF, Hertzberg AM (1985)
Bouaziz, O. (2023)
Breiman, L. (2001)
Cook, R. J., & Lawless, J. (2007)
Devaux, A, et. Al (2023)
Feurer, M., & Hutter, F. (2019)
Harrell Jr, F. E., Lee, K. L., & Mark, D. B. (1996)
Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009)
Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008)
Kaplan, E. L., & Meier, P. (1958)
Kim, S., Schaubel, D. E., & McCullough, K. P. (2018)
Kvamme, H., & Borgan, Ø. (2019)
Murris, J., Charles-Nelson, A., Lavenu, A., & Katsahian, S. (2022)
Nelson, W. B. (2003)
Therneau, T., Grambsch, P., & Fleming, T. (1990)