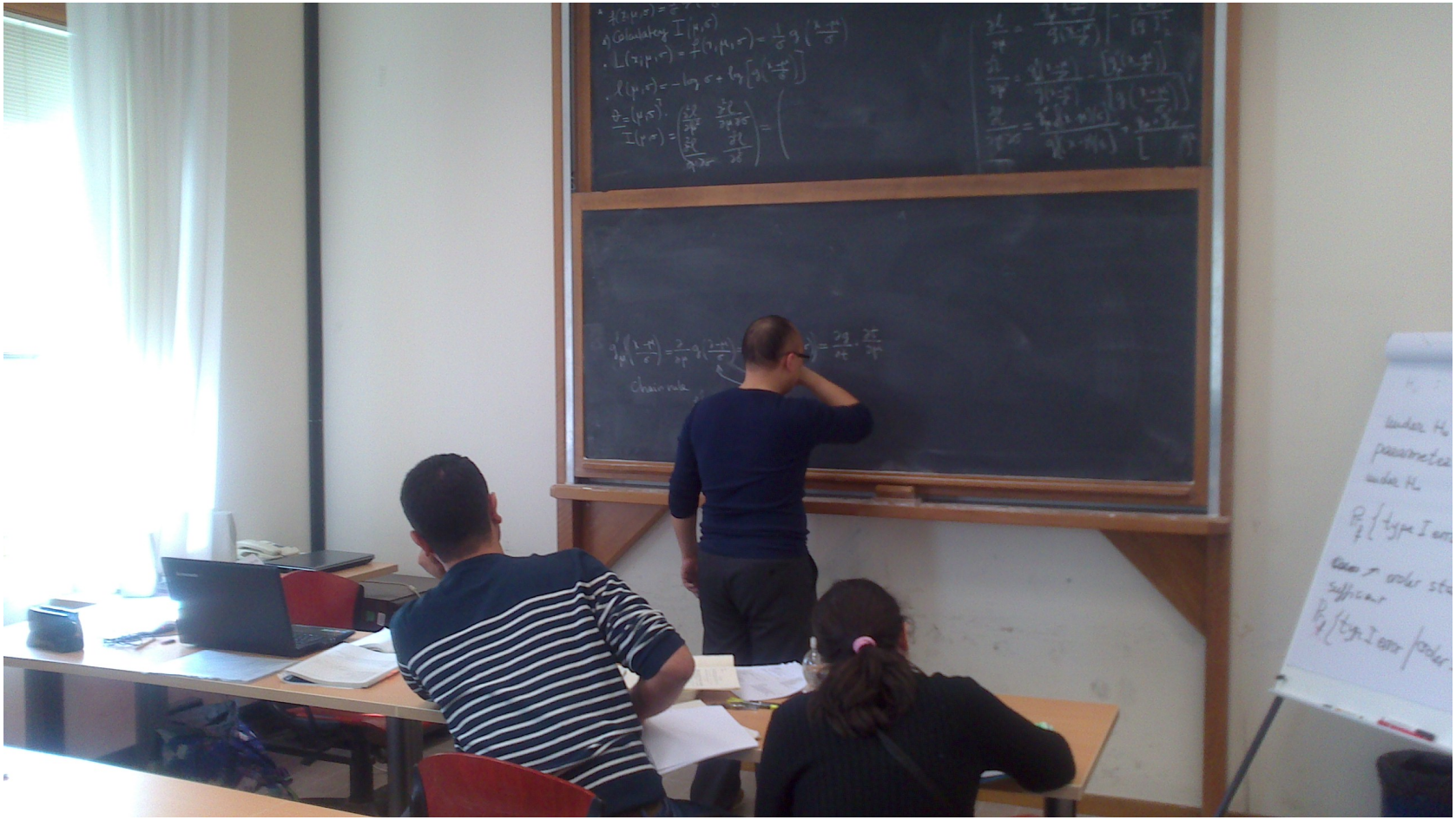
Statistical methods for the dynamic prediction of survival in settings with numerous longitudinal predictors

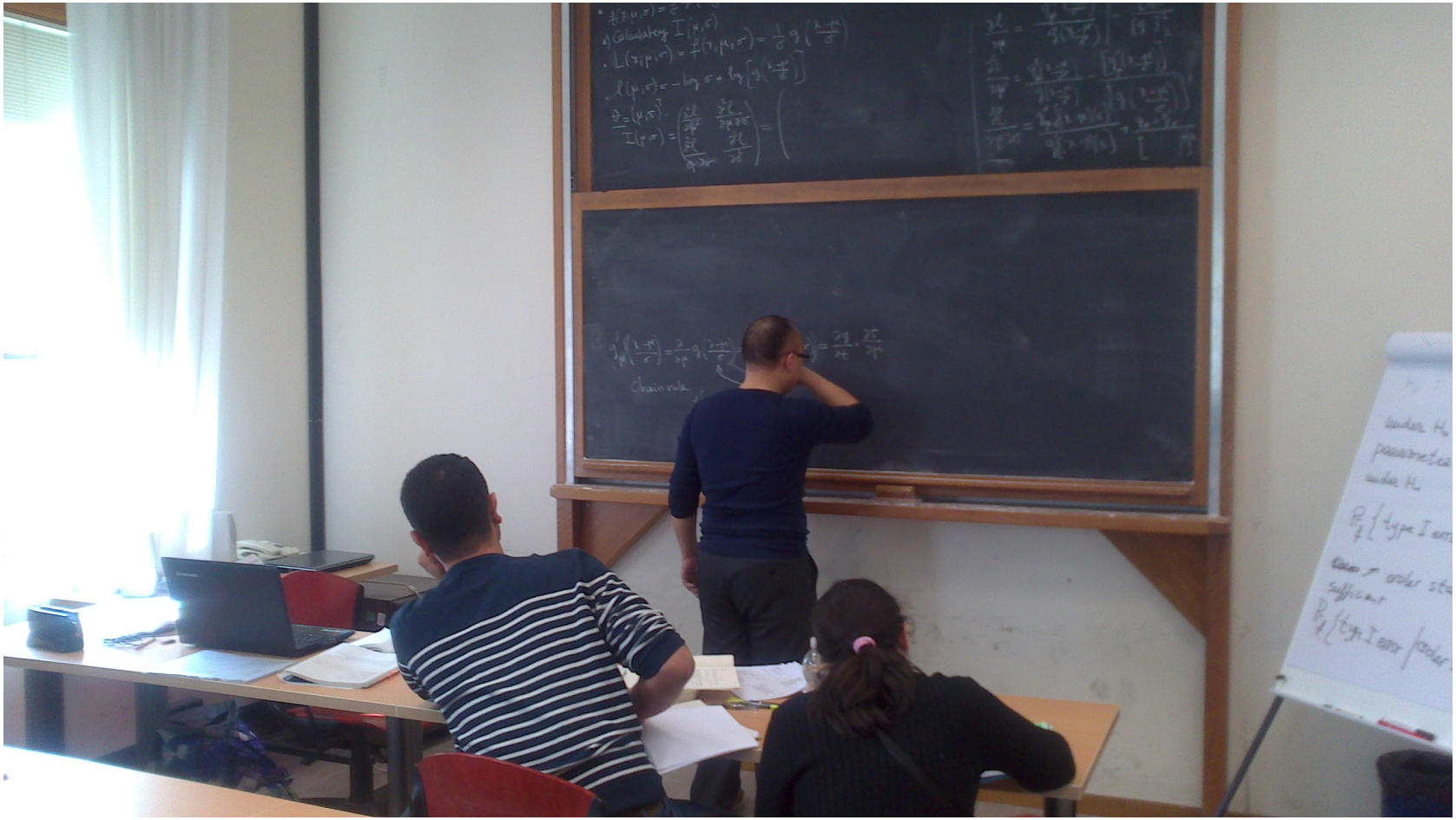
Mirko Signorelli
: mirkosignorelli.github.io
: @signormirko

Mathematical Institute
Leiden University

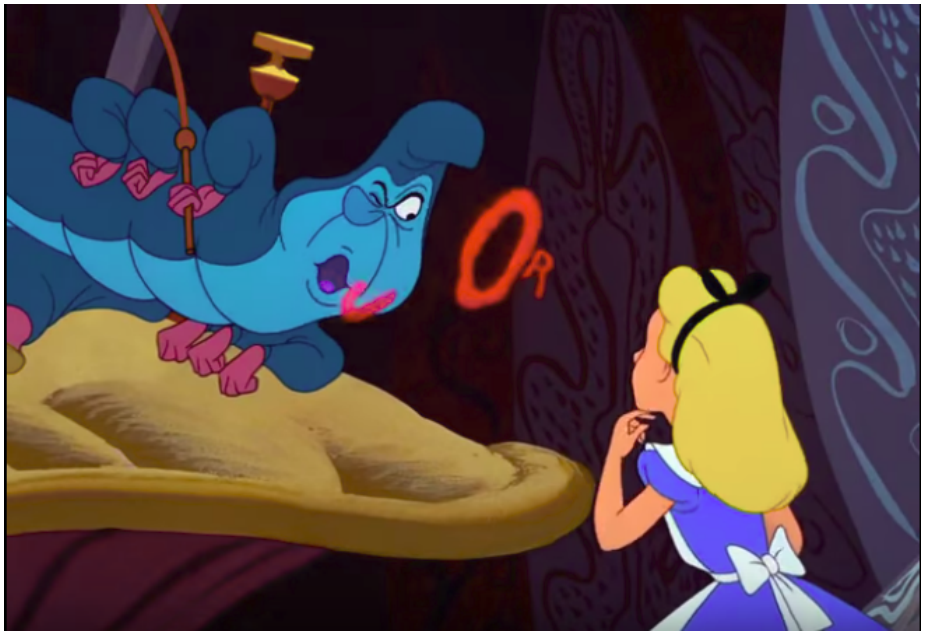
April 3, 2024
Dipartimento di Scienze Statistiche, UniPD

Universiteit Leiden





- Photo from 2014
- Last time in Padova: May 22, 2017


OR

Who are you?

- Assistant prof. in Statistics at the Mathematical Institute of Leiden University (NL)
- PhD between University of Padova (IT) and of Groningen (NL)
- Research interests:

Who are you?

- Assistant prof. in Statistics at the Mathematical Institute of Leiden University (NL)
- PhD between University of Padova (IT) and of Groningen (NL)
- Research interests:
  - statistical modelling of networks (graphs & hypergraphs)

Who are you?

- Assistant prof. in Statistics at the Mathematical Institute of Leiden University (NL)
- PhD between University of Padova (IT) and of Groningen (NL)
- Research interests:
  - statistical modelling of networks (graphs & hypergraphs)

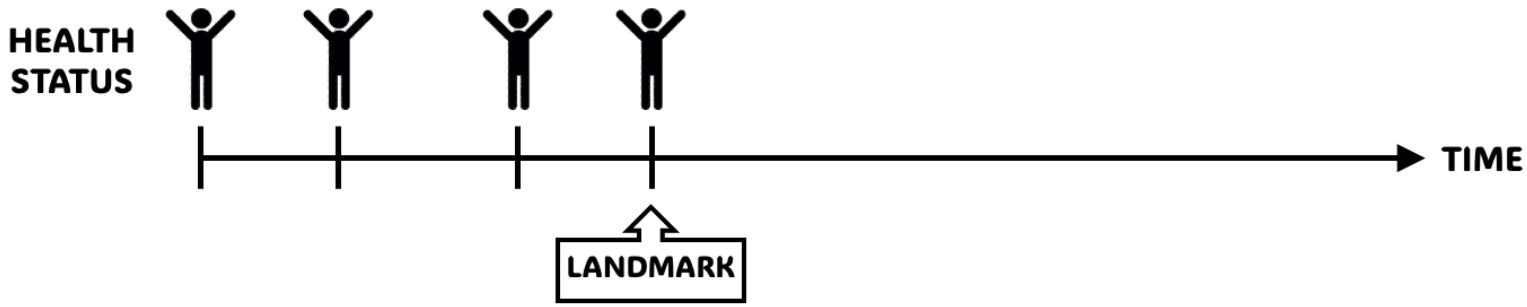
VENERDI' 5 APRILE 2024 ORE 9.00 IN AULA SC140
COMPLESSO SANTA CATERINA VIA CESARE BATTISTI, 241

Orario	N.	Corso di Laurea	Matricola	COGNOME	TITOLO DELLA TESI
		Lauree Magistrali			
09:00	1	SS1736	2053005	CALORE ALBERTO	Metodi di bilanciamento in presenza di trattamenti multipli: MARMoT e Template Matching a confronto
09:30	2	SS1736	2058300	DRIUSSO EUGENIA	Statistical modelling of time-stamped hypergraphs: a model-based clustering approach
10:00	3	SS1736	2053208	FRAULINI ENRICO	Effetto delle differenze di reddito sulle elezioni presidenziali statunitensi

Who are you?

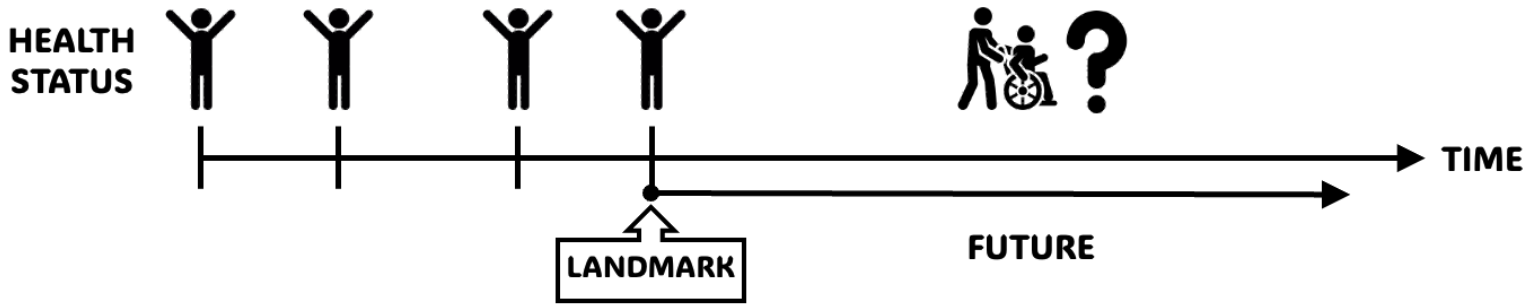
- Assistant prof. in Statistics at the Mathematical Institute of Leiden University (NL)
- PhD between University of Padova (IT) and of Groningen (NL)
- Research interests:
  - statistical modelling of networks (graphs & hypergraphs)
  - longitudinal data analysis
  - survival analysis
  - biomedical applications

Dynamic prediction 101



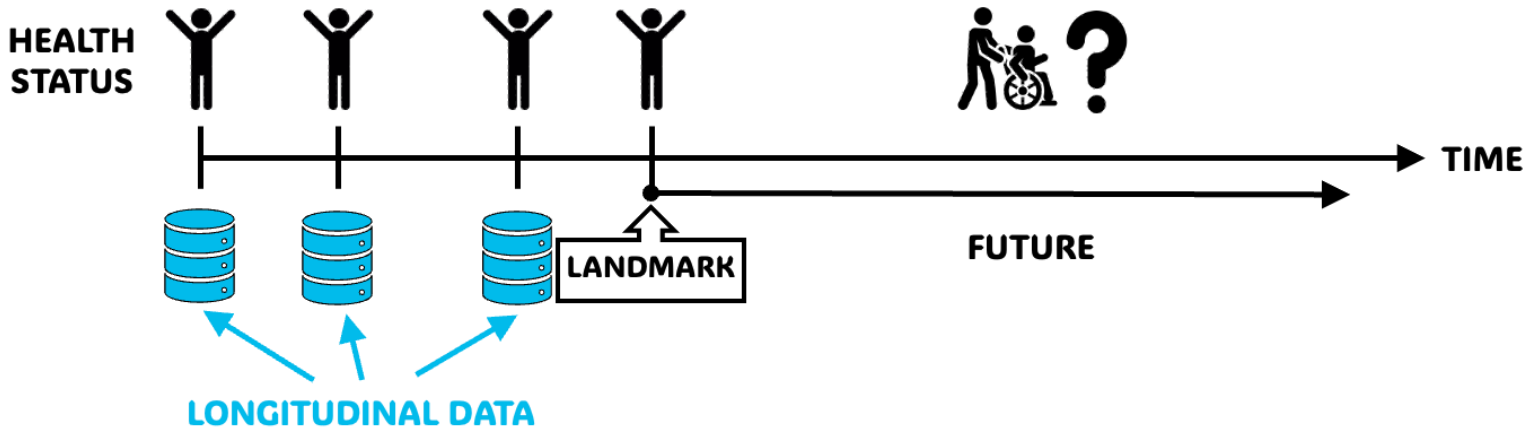
→ example

Dynamic prediction 101



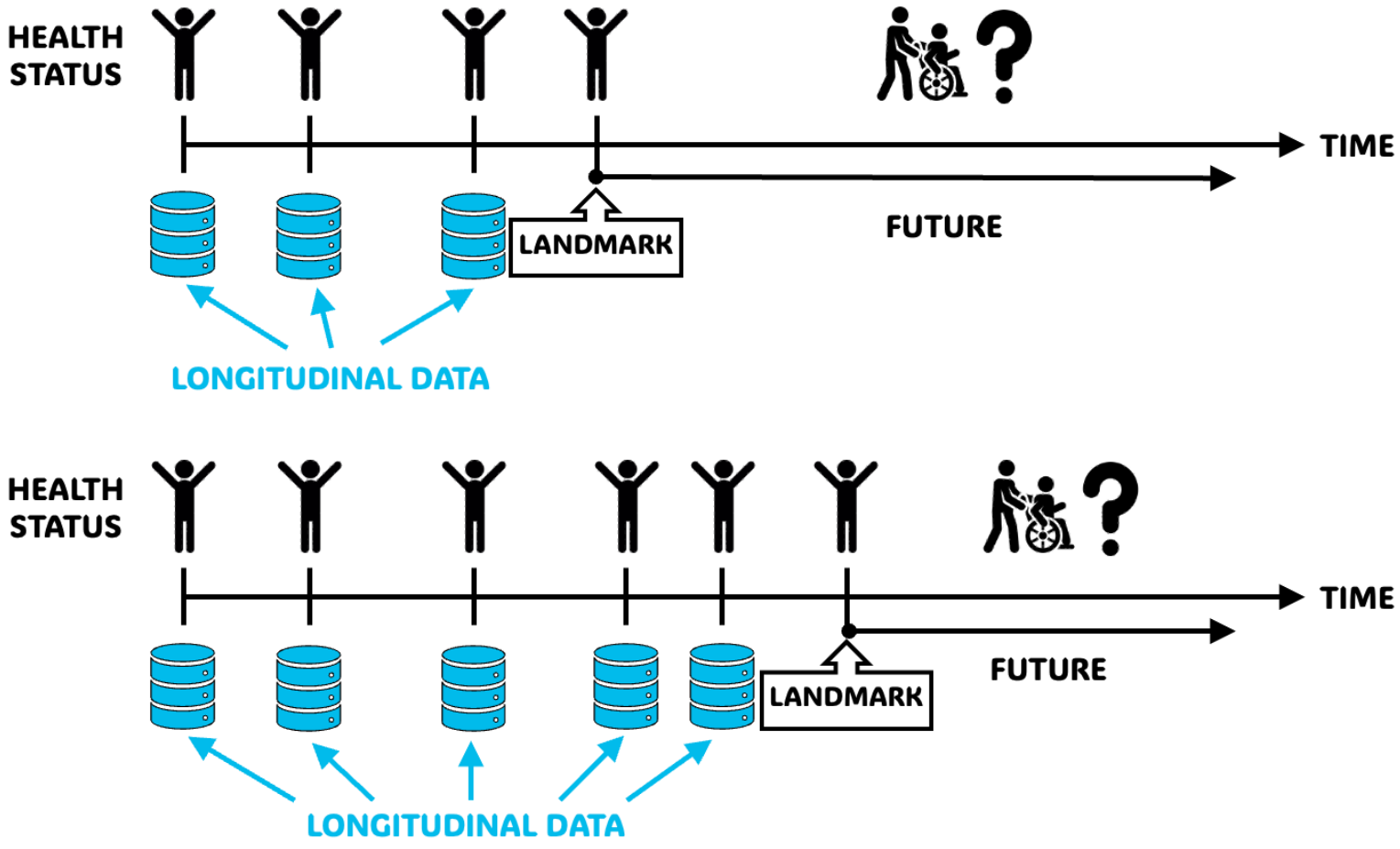
→ example

Dynamic prediction 101



→ example

Dynamic prediction 101



→ example

The goal(s)

Goals of dynamic prediction:

- predict future survival $S(t|\ell_1) = P(T > t|\, T > \ell_1),\ t > \ell_1$, using repeated measurements over $[0, \ell_1]$
- update predictions when newer information becomes available, i.e. update $S(t|\ell_2)$ given repeated measurements over $[0, \ell_2]$, $t > \ell_2 > \ell_1$

The problem

- Traditional methods for dynamic prediction:
  - joint models: very computationally-intensive. Can't usually be estimated with more than 3-5 longitudinal predictors!
  - landmarking with LOCF[1]: no modelling of the longitudinal trajectories + no measurement error correction (important for biomarkers)
- Problem: nowadays, longitudinal studies can comprise tens, hundreds, or even thousands longitudinal predictors ("biomarkers")
- How to do dynamic prediction with "many" longitudinal predictors?

[1] LOCF = Last Observation Carried Forward

Recent methodological solutions

Several solutions proposed over the last 5 years:

1. Li & Luo (2019): MFPCCox
2. Signorelli et al. (2021)[2]: Penalized Regression Calibration (PRC)
3. Lin et al. (2021): Functional Random Survival Forest (FunRSF)
4. Devaux et al. (2023): DynForest

[2] but see Signorelli (2023) for a better explanation of the dynamic prediction setting

Modelling approaches

- Two-step modelling:
  1. model trajectories of longitudinal predictors over $[0, \ell]$
  2. use summaries of longitudinal predictors to predict $S(t|\ell),\ t \geq \ell$

		Longitudinal covariates	
		Multivariate Functional PCA	Mixed Effects Models
Survival outcome	Cox PH model	**MFPCCox** (Li and Luo, 2019)	**Penalized Regression Calibration** (Signorelli et al., 2021)
	Random survival forest	**Functional Random Survival Forest** (Lin et al., 2021)	**DynForest** (Devaux et al., 2023)

Notation

Covariates:

- P baseline covariates $x_i = (x_{1i}, ..., x_{Pi})$ measured at $t_{i1} = 0$
- Q longitudinal covariates measured at $t_{i1}, t_{i2}, ..., t_{im_i}$
- $y_{qij} = y_{qi}(t_{ij})$

Survival outcome:

- T_i time-to-event outcome for subject i
- δ_i event indicator:
  - $\delta_i = 1$: event observed at $T_i = t_i$
  - $\delta_i = 0$: right-censoring at $T_i = t_i$

The 4 bricks

		Longitudinal covariates	
		Multivariate Functional PCA	Mixed Effects Models
Survival outcome	Cox PH model	**MFPCCox** (Li and Luo, 2019)	**Penalized Regression Calibration** (Signorelli et al., 2021)
	Random survival forest	**Functional Random Survival Forest** (Lin et al., 2021)	**DynForest** (Devaux et al., 2023)

- We need 4 "bricks":
  1. MFPCA / mixed-effects models
  2. Cox model / random survival forest

Building block 1: MFPCA

- MFPCA (Happ & Greven, 2018) decomposition:

$$y_{qij} = y_{qi}(t_{ij}) \approx \mu_q(t_{ij}) + \sum_{k=1}^{K} \rho_{ki}\psi_{kq}(t_{ij}),\ i \in \mathcal{I}(\ell),\ t_{ij} \leq \ell, \qquad (1)$$

where:

- $\psi_k(t) = (\psi_{k1}(t), ..., \psi_{kQ}(t))$ are Q-variate orthonormal eigenfunctions
- ρ_{ki} are subject-specific MFPCA scores (shared across $Y_1, ..., Y_Q$)
- eigenfunctions are ordered by decreasing percentage of variance explained (PVE)
- K chosen so that eigenfunctions explain a certain PVE of the original variables (usually: 90 / 95%)

Building block 2: linear mixed models (LMMs)

- Linear mixed model (LMM):

$$y_{qij} = y_{qi}(t_{ij}) = W_{qi}(t_{ij})\beta_q + Z_{qi}(t_{ij})u_{qi} + \varepsilon_{qi},\ i \in \mathcal{I}(\ell),\ t_{ij} \leq \ell, \quad (2)$$

where:

- $W_{qi}(t_{ij}), Z_{qi}(t_{ij})$ are design matrices
- $u_{qi} \sim N(0, \Sigma_q) \rightarrow$ random effects
- $\varepsilon_{qi} \sim N(0, \sigma_q^2)$
- $\beta_q, \Sigma_q, \sigma_q^2 \rightarrow$ fixed effects

Building block 3: Cox proportional-hazards model

- Cox model (Cox, 1972):

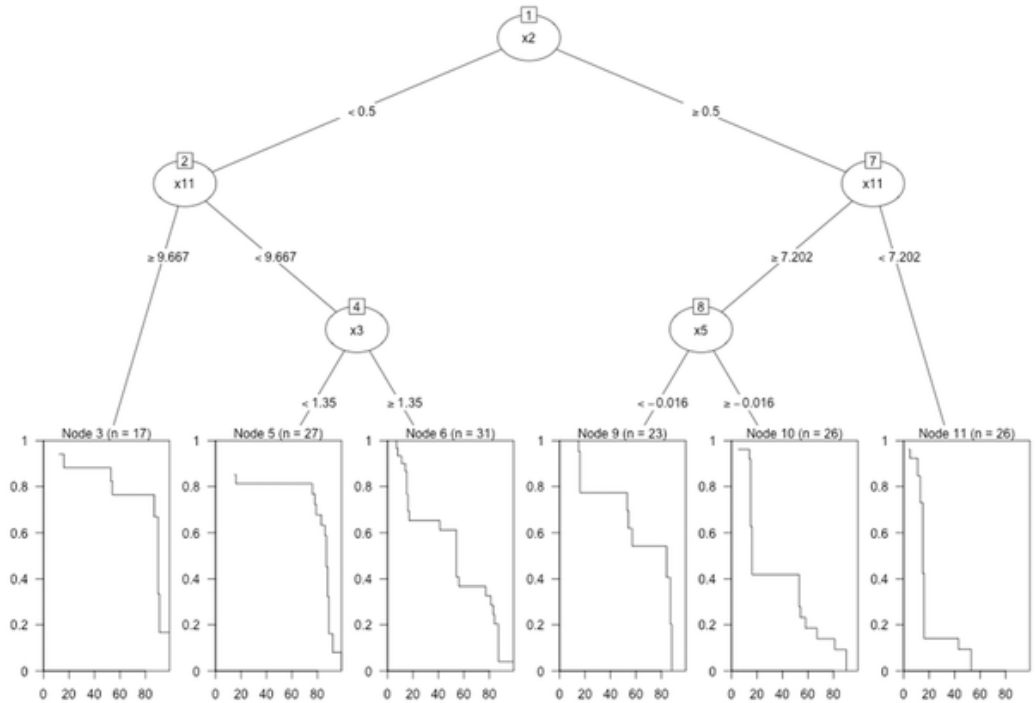
$$h_i(t) = h_0(t) \exp\left(\sum_j \gamma_j z_j\right), \tag{3}$$

where:

- $h_i(t) = \lim_{h \to 0^+} P\left(T_i \in [t, t+h)\right)$ is the hazard function
- $h_0(t)$ is the baseline hazard
- z_j covariate with regression coefficient γ_j

Building block 4: random survival forest

- Draw B bootstrap samples from data
- For $b = 1, ..., B$:
  - specify tuning parameters, e.g. number of candidate covariates for node splitting, minimum node size, minimum number of events...
  - build a survival tree



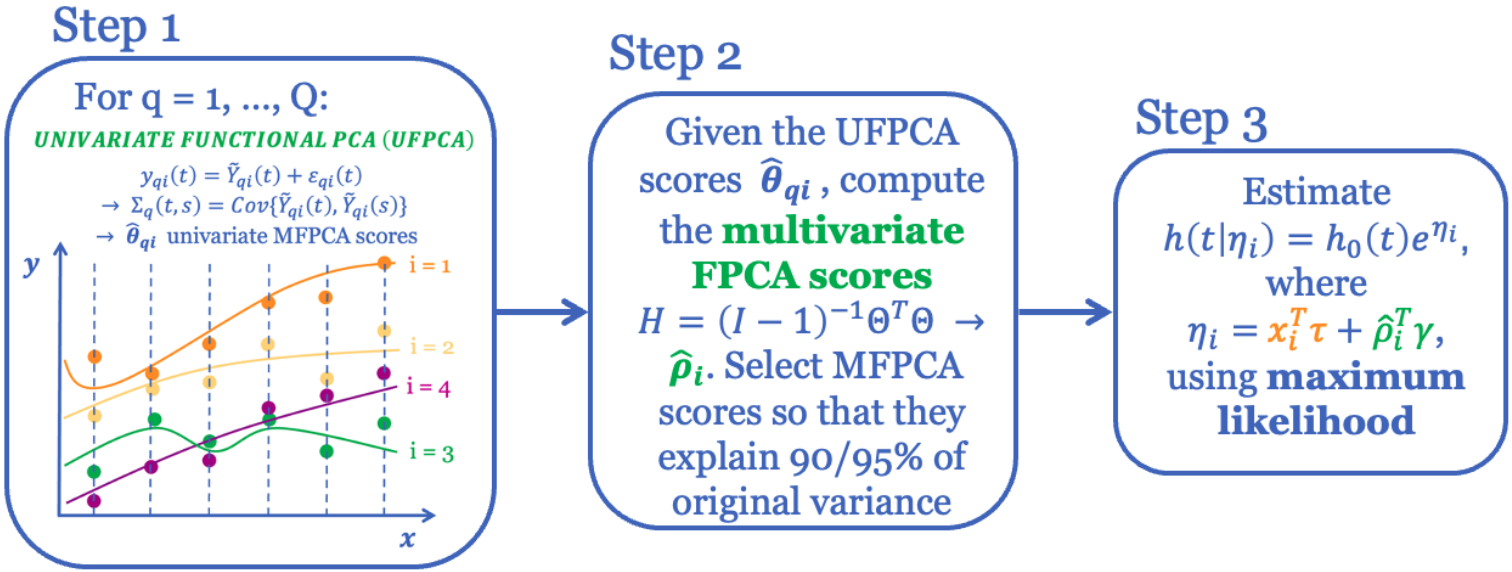
Credits: Wetten et al. (2021), PLOS ONE, 16(5), e0250963

Building block 4: random survival forest

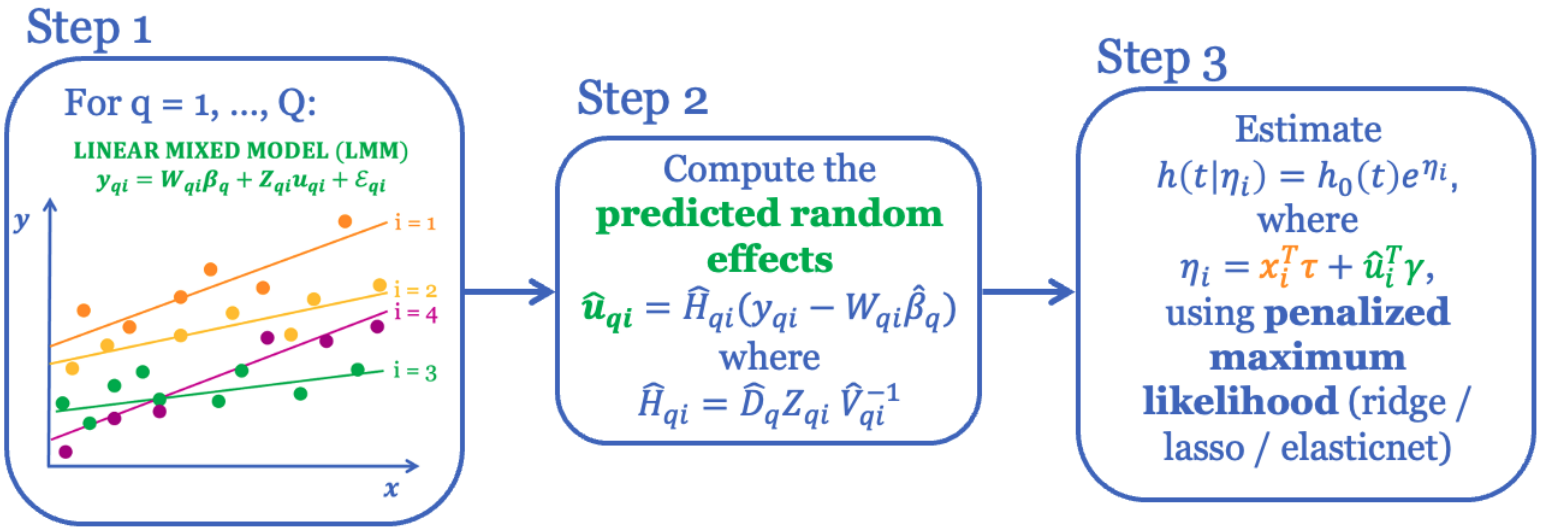
- Draw B bootstrap samples from data
- For $b = 1, ..., B$:
  - specify tuning parameters, e.g. number of candidate covariates for node splitting, minimum node size, minimum number of events...
  - build a survival tree
  - predicted survival for subject i with covariates z_i given by cumulative hazard function (CHF) of the terminal node where i ends, $\hat{H}_b(t|x_i)$ ($\rightarrow \hat{S}(t) = \exp \hat{H}(t)$)
- Average CHF predictions across trees:

$$\hat{H}(t|z_i) = \frac{1}{B} \sum_{b=1}^{B} \hat{H}_b(t|x_i) \tag{4}$$

MFPCCox (Li & Luo, 2019)

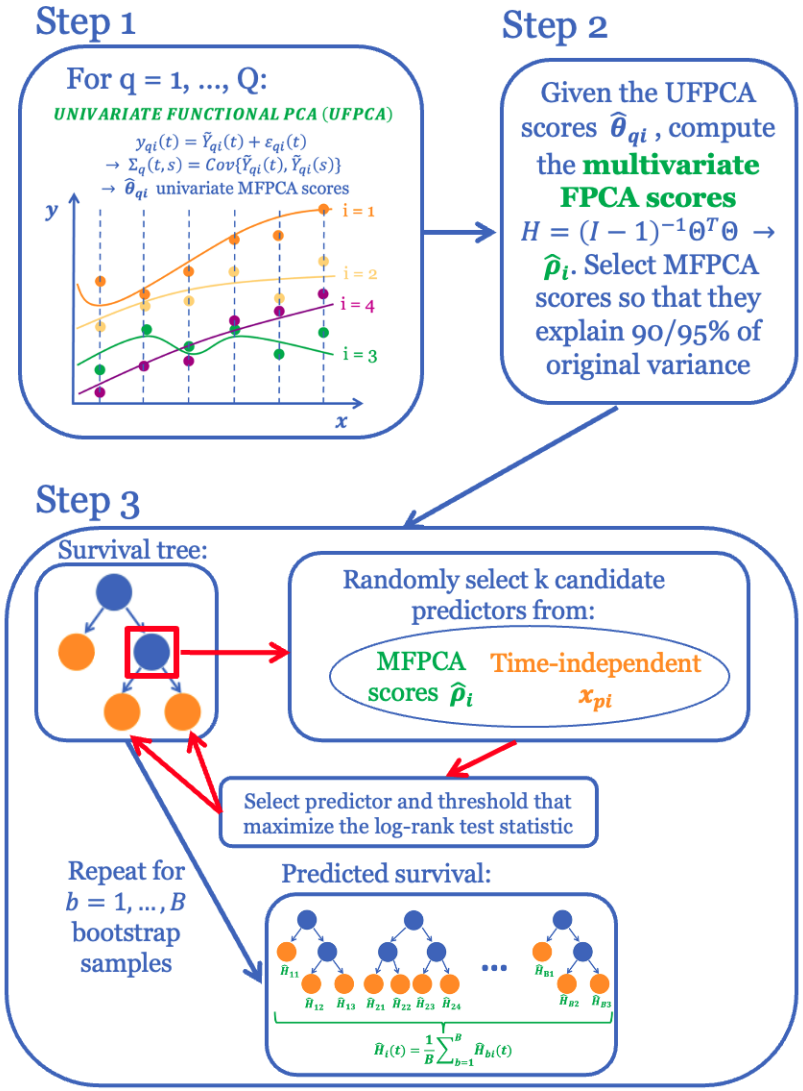


Penalized Regression Calibration (PRC, Signorelli et al. (2021))

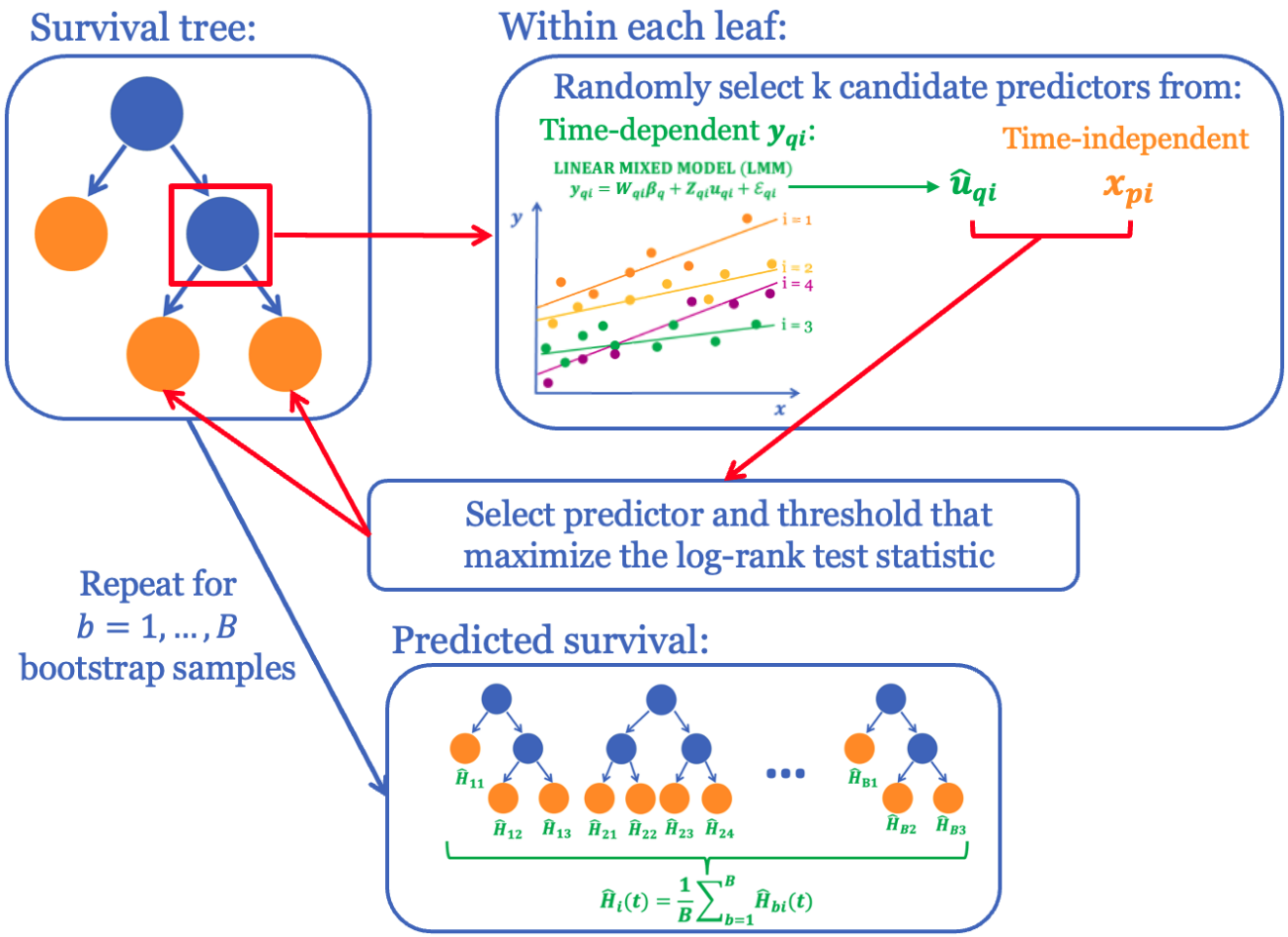


- Steps 1-2: multivariate version with MLPMM possible

Functional Survival Random Forest (FunRSF, Lin et al. (2021))



DynForest (Devaux et al., 2023)



Motivation

- Methods proposed quite recently: between 2019 and 2023
- Very little knowledge about their performance with real data from longitudinal studies
- Let's compare them!

Preprint (Signorelli & Retif, 2024)

Statistics > Methodology

[Submitted on 21 Mar 2024]

An empirical appraisal of methods for the dynamic prediction of survival with numerous longitudinal predictors

Signorelli Mirko, Sophie Retif

Recently, the increasing availability of repeated measurements in biomedical studies has motivated the development of several statistical methods for the dynamic prediction of survival in settings where a large (potentially high-dimensional) number of longitudinal covariates is available. These methods differ in both how they model the longitudinal covariates trajectories, and how they specify the relationship between the longitudinal covariates and the survival outcome. Because these methods are still quite new, little is known about their applicability, limitations and performance when applied to real-world data.
To investigate these questions, we present a comparison of the predictive performance of the aforementioned methods and two simpler prediction approaches to three datasets that differ in terms of outcome type, sample size, number of longitudinal covariates and length of follow-up. We discuss how different modelling choices can have an impact on the possibility to accommodate unbalanced study designs and on computing time, and compare the predictive performance of the different approaches using a range of performance measures and landmark times.

Subjects: **Methodology (stat.ME)**; Applications (stat.AP)
Cite as: arXiv:2403.14336 **[stat.ME]**
(or arXiv:2403.14336v1 **[stat.ME]** for this version)

Longitudinal studies

- We considered data from three longitudinal studies:

ROSMAP

- Event: Alzheimer's Disease diagnosis
- n = 3293
- 5 baseline covariates
- 30 longitudinal covariates
- Follow-up: [1, 29] years

ADNI

- Event: diagnosis of dementia
- n = 1643
- 5 baseline covariates
- 21 longitudinal covariates
- Follow-up: [0, 15.5] years

PBC2

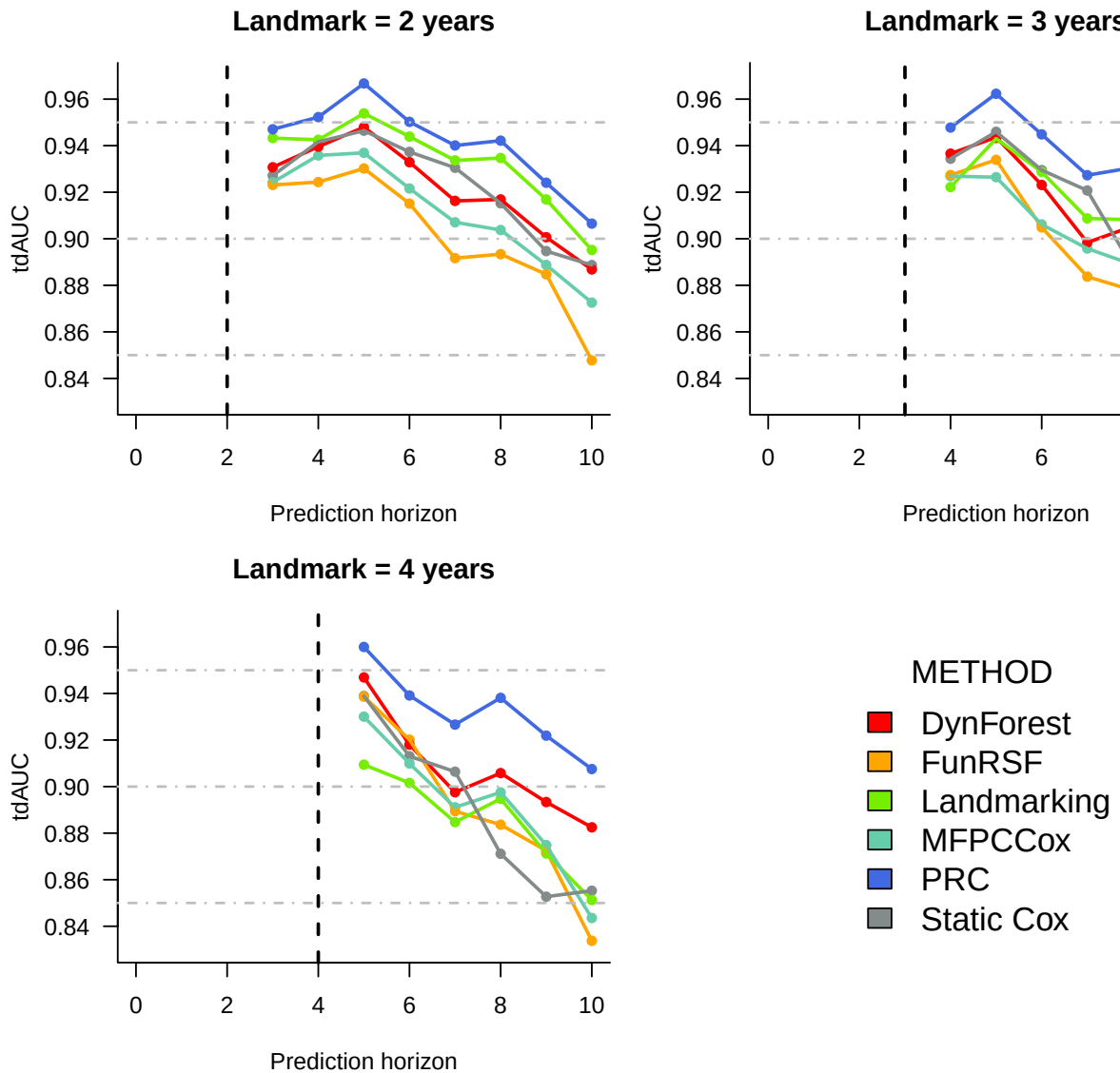
- Event: death (primary biliary cirrhosis trial)
- n = 312
- 3 baseline covariates
- 8 longitudinal covariates
- Follow-up: [0, 14] years

- Methods included: MFPCCox, PRC, FunRSF, DynForest + static Cox + LOCF landmarking
- Performance evaluated at multiple landmark times
- Performance measures: C index, tdAUC, Brier score
- 10-fold cross-validation, repeated 10 times

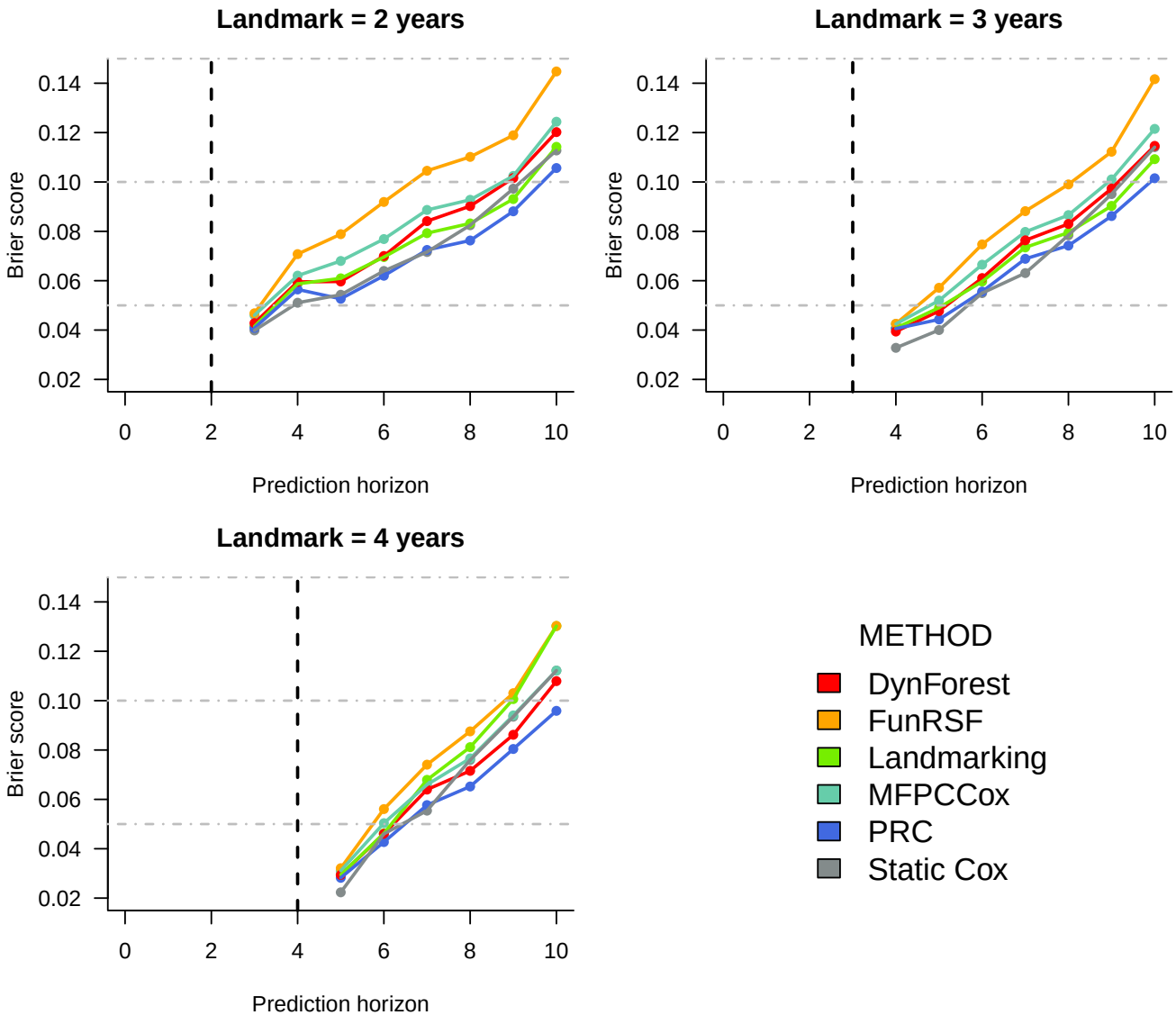
ADNI dataset: C index

	Landmark		
Method	2	3	4
Static Cox	0.901 (0.002)	0.885 (0.006)	0.856 (0.008)
Landmarking	0.906 (0.001)	0.89 (0.004)	0.855 (0.009)
MFPCCox	0.889 (0.003)	0.872 (0.009)	0.859 (0.008)
PRC	0.913 (0.001)	0.908 (0.003)	0.904 (0.003)
FunRSF	0.873 (0.004)	0.858 (0.011)	0.845 (0.012)
DynForest	0.891 (0.003)	0.883 (0.005)	0.871 (0.011)

ADNI dataset: time-dependent AUC



ADNI dataset: Brier score



Results overview

- PRC, landmarking, DynForest $>$ MFPCCox, static Cox, FunRSF
- Methods that use LMMs $>$ methods using MFPCA
- Conditionally on method used to model longitudinal predictors (MFPCA / LMMs), methods that use Cox model $>$ methods that use RSF
- LOCF landmarking often second / third best model
- Relative performance of landmarking and static Cox worsens with higher landmark & horizon times

More details: arXiv:2403.14336

Limitations

- MFPCA-based methods: regular measurement grid required $\rightarrow$ unrealistic & unflexible
- LMM-based methods: using GLMMs would allow for more modelling flexibility
- Methods using RSF: need to choose value of multiple tuning parameters
- Competing risks: only in DynForest
- Interval censoring: none of the methods
- Software
  - MFPCCox, FunRSF: no software implementation!
  - PRC $\rightarrow$ `pencal`, DynForest $\rightarrow$ `DynForest`

Where to find the package

- PRC implemented in the `R` package `pencal`
- Available on CRAN:

`pencal: Penalized Regression Calibration (PRC) for the Dynamic Prediction of Survival`

Computes penalized regression calibration (PRC), a statistical method for the dynamic prediction of survival when many longitudinal predictors are available. PRC is described in Signorelli et al. (2021) <doi:10.1002/sim.9178> and Signorelli (2023) <doi:10.48550/arXiv.2309.15600>.

Version:	2.1.1
Depends:	R (≥ 4.1.0)
Imports:	doParallel, dplyr, foreach, glmnet, lcmm, magic, MASS, Matrix, methods, nlme, purrr, riskRegression, stats, survcomp, survival, survivalROC
Suggests:	knitr, ptmixed, rmarkdown, survminer
Published:	2023-10-27
Author:	Mirko Signorelli [aut, cre, cph], Pietro Spitali [ctb], Roula Tsonaka [ctb], Barbara Vreede [ctb]
Maintainer:	Mirko Signorelli <msignorelli.rpackages at gmail.com>
License:	GPL (≥ 3)
URL:	https://mirkosignorelli.github.io/r
NeedsCompilation:	no
Citation:	pencal citation info
Materials:	NEWS
CRAN checks:	pencal results

`Documentation:`

Reference manual:	pencal.pdf
Vignettes:	pencal: an R Package for the Dynamic Prediction of Survival with Many Longitudinal Predictors

Example dataset

- Data from the PBC2 clinical trial (1974-1984)
  - n = 312, P = 3 baseline and Q = 7 longitudinal predictors
  - Outcome: time to death
  - Follow-up up to 14.3 years

```
library(pencal)
data(pbc2data)
sdata = pbc2data$baselineInfo
ldata = pbc2data$longitudinalInfo
```

Data preparation

- Let's choose $\ell = 2$ as landmark:

```
# remove subjects with event / censoring before landmark
lmark = 2
sdata = subset(sdata, time > lmark)
ldata = subset(ldata, id %in% sdata$id)

# remove repeated measurements taken after landmark
ldata = subset(ldata, futime <= lmark)
```

- We log-transform some highly-skewed predictors:

```
ldata$logSerBil = log(ldata$serBilir)
ldata$logSerChol = log(ldata$serChol)
ldata$logAlk = log(ldata$alkaline)
ldata$logSGOT = log(ldata$SGOT)
ldata$logProthr = log(ldata$prothrombin)
```

Inputs

1. A dataset (`ldata`) with the longitudinal covariates measured up to the landmark time:

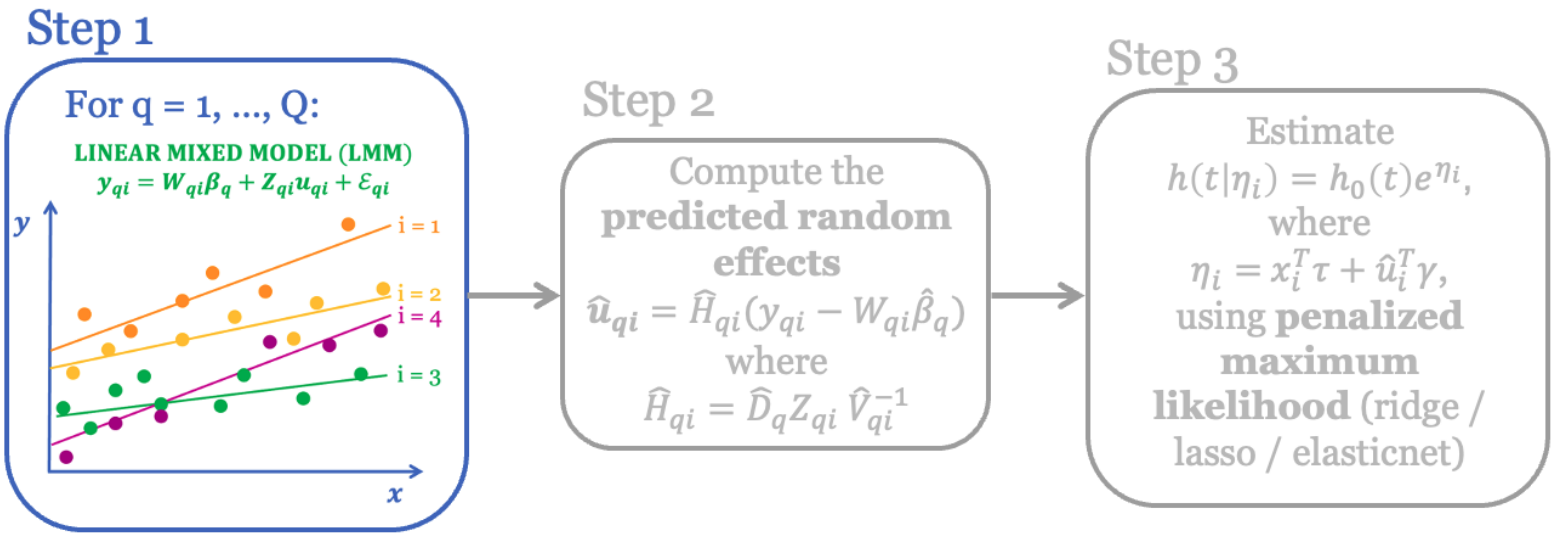
```
##    id   age fuptime logSerBil logSerChol albumin logAlk
## 3   2 56.45    0.00      0.10       5.71    4.14   8.91
## 4   2 56.95    0.50     -0.22         NA    3.60   7.65
## 5   2 57.45    1.00      0.00         NA    3.55   7.44
## 16  4 54.74    0.00      0.59       5.50    2.54   8.72
## 17  4 55.26    0.51      0.47         NA    2.88   7.07
## 18  4 55.76    1.02      0.53         NA    2.80   7.05
## 19  4 56.74    2.00      1.16         NA    2.92   7.07
##    logSGOT platelets logProthr
## 3     4.73       221      2.36
## 4     4.94       188      2.40
## 5     4.97       161      2.45
## 16    4.10       183      2.33
## 17    5.13       240      2.94
## 18    5.11       251      2.45
## 19    5.12       220      2.38
```

Inputs

2. A dataset (`sdata`) with the survival outcome, and baseline covariates:

```
##    id      time event baselineAge    sex treatment
## 3   2 14.152338     0    56.44782 female D-penicil
## 12  3  2.770781     1    70.07447   male D-penicil
## 16  4  5.270507     1    54.74209 female D-penicil
## 23  5  4.120578     0    38.10645 female   placebo
## 29  6  6.853028     1    66.26054 female   placebo
## 35  7  6.847552     0    55.53609 female   placebo
```

Step 1: estimating the LMMs



Step 1: estimating the LMMs

```
long_covs = c('logSerBil', 'logSerChol', 'albumin',
              'logAlk', 'logSGOT', 'platelets',
              'logProthr')

step1 = fit_lmms(y.names = long_covs,
                 fixefs = ~ age, ranefs = ~ age | id,
                 long.data = ldata, surv.data = sdata,
                 t.from.base = fuptime)
```

Extracting output from the fitted LMMs

```
summary(step1, yname = 'logSerBil', what = 'betas') |> round(6)
```

```
## (Intercept)          age
##    0.518320   -0.001045
```

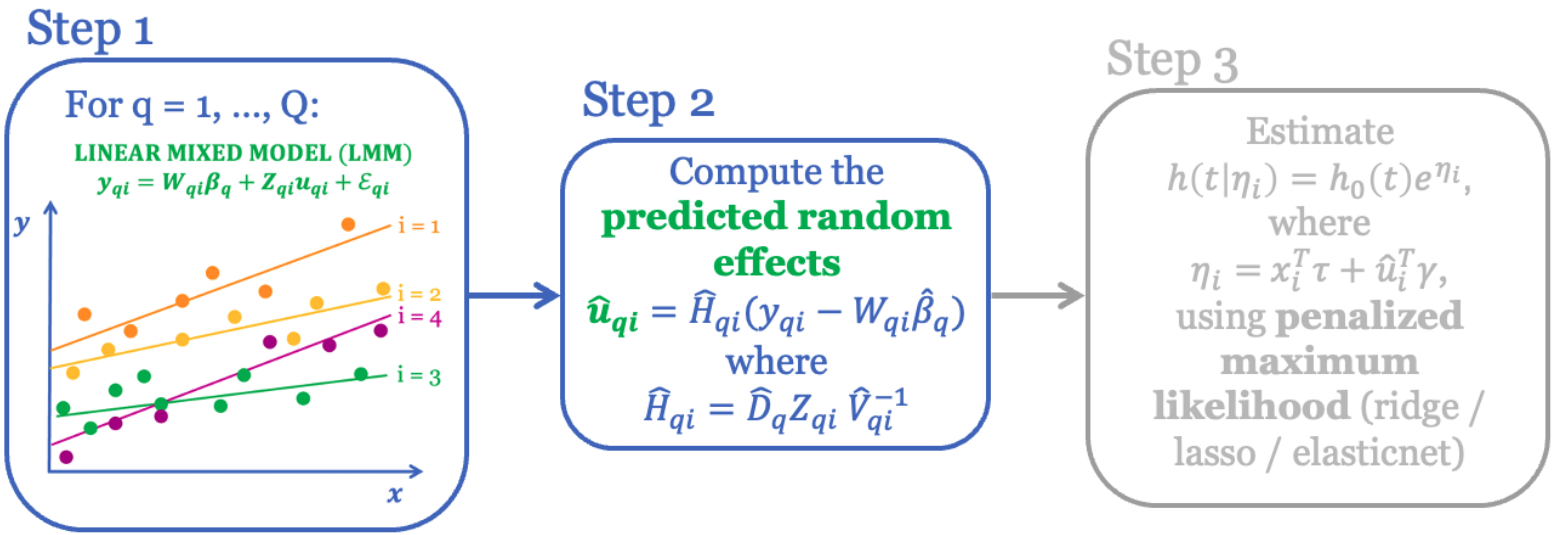
```
summary(step1, yname = 'logSerBil', what = 'tTable') |> round(4)
```

```
##                Value Std.Error  DF t-value p-value
## (Intercept)  0.5183    0.2788 566  1.8590  0.0636
## age         -0.0010    0.0055 566 -0.1884  0.8506
```

```
summary(step1, yname = 'logSerBil', what = 'variances')
```

```
## id = pdLogChol(age)
##                Variance     StdDev      Corr
## (Intercept) 7.332118e-01 0.856277849 (Intr)
## age         4.731627e-05 0.006878682 0.103
## Residual    1.437622e-01 0.379159888
```

Step 2: computing the predicted random effects



Step 2: computing the predicted random effects

```
step2 = summarize_lmms(step1)
```

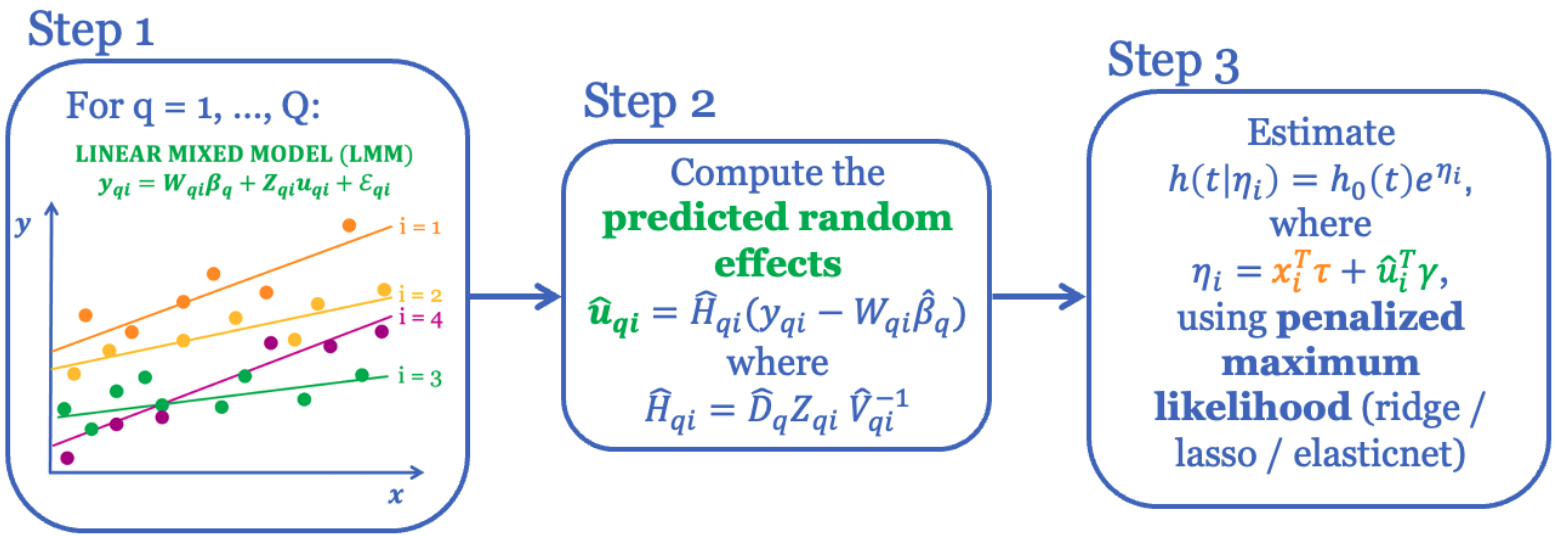
- Handy: `summarize_lmms` automatically inherits relevant arguments from `fit_lmms` ☺

Step 2: sample output

```
round(step2$ranef.orig[1:5, 1:6], 6)
```

```
##   logSerBil_b_int logSerBil_b_age logSerChol_b_int
## 2       -0.382988       -0.001661        -0.071154
## 3       -0.117107       -0.000584        -0.598453
## 4        0.168600        0.000922        -0.370434
## 5        0.380035        0.001170        -0.291031
## 6       -0.473763       -0.002305        -0.248214
##   logSerChol_b_age albumin_b_int albumin_b_age
## 2         0.000660      0.179725       3.0e-06
## 3         0.004916      0.018124       1.0e-06
## 4         0.003468     -0.529776      -7.0e-06
## 5         0.002886     -0.148329       8.0e-06
## 6         0.002136      0.292353       1.7e-05
```

Step 3: estimate the penalized Cox model



Step 3: estimate the penalized Cox model

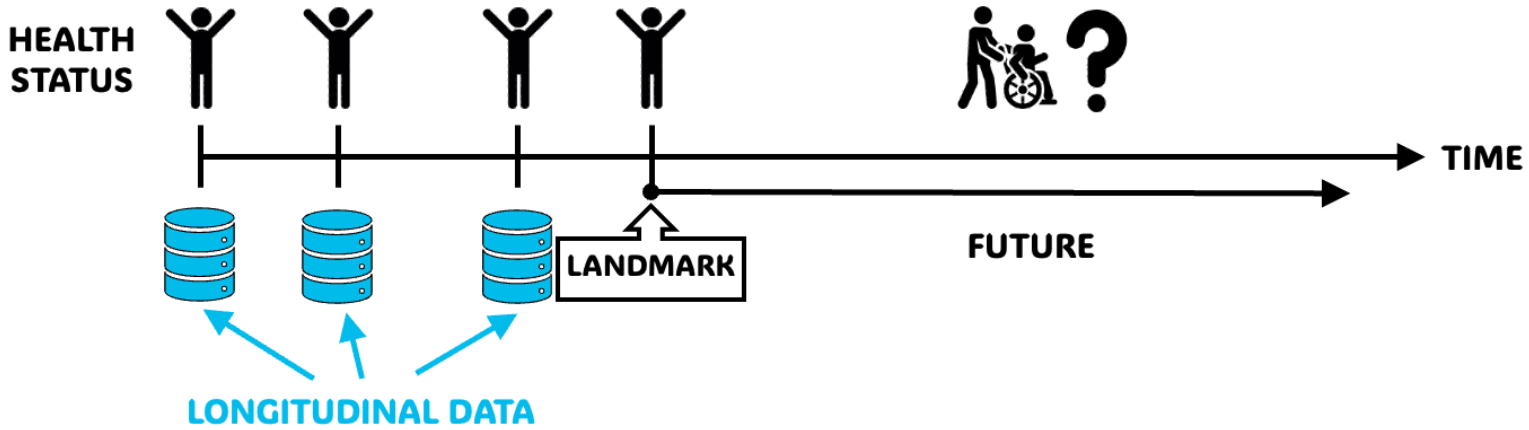
```
step3 = fit_prclmm(step2, surv.data = sdata,
      baseline.covs = ~ baselineAge + sex + treatment,
      penalty = 'ridge', standardize = T)
```

Step 3: fitted model

```
summary(step3)
```

```
## Fitted model: PRC-LMM
## Penalty function used: ridge
## Tuning parameters:
##      lambda alpha
## 1 0.2126761     0
## Sample size: 278
## Number of events: 107
## Bootstrap optimism correction: not computed
## Penalized likelihood estimates (rounded to 4 digits):
##   baselineAge sexfemale treatmentD-penicil logSerBil_b_int
## 1      0.0476   -0.2872            -0.0157          0.4341
##   logSerBil_b_age logSerChol_b_int logSerChol_b_age
## 1        111.3935           0.0986         -10.5311
##   albumin_b_int albumin_b_age logAlk_b_int logAlk_b_age
## 1       -1.1361      23070.92       0.0874     -12.5617
##   logSGOT_b_int logSGOT_b_age platelets_b_int
## 1         0.238       272.246         -0.0011
##   platelets_b_age logProthr_b_int logProthr_b_age
## 1         -0.2046          2.8114       -573.3093
```

Back to our goal: predicting survival



Prediction of survival

```
Shat = survpred_prclmm(step1, step2, step3, times = 3:5)
```

- This will compute $\hat{S}(t|2),\ t = 3, 4, 5$:

```
head(Shat$predicted_survival, 4)
```

```
##   id      S(3)      S(4)      S(5)
## 2  2 0.9398512 0.8867592 0.8329966
## 3  3 0.8555809 0.7392053 0.6316391
## 4  4 0.8138498 0.6709525 0.5451302
## 5  5 0.9460431 0.8981124 0.8492655
```

- Prediction for new subjects? Possible through additional arguments `new.longdata` and `new.basecovs`
- Evaluation of predictive performance: see → Appendix

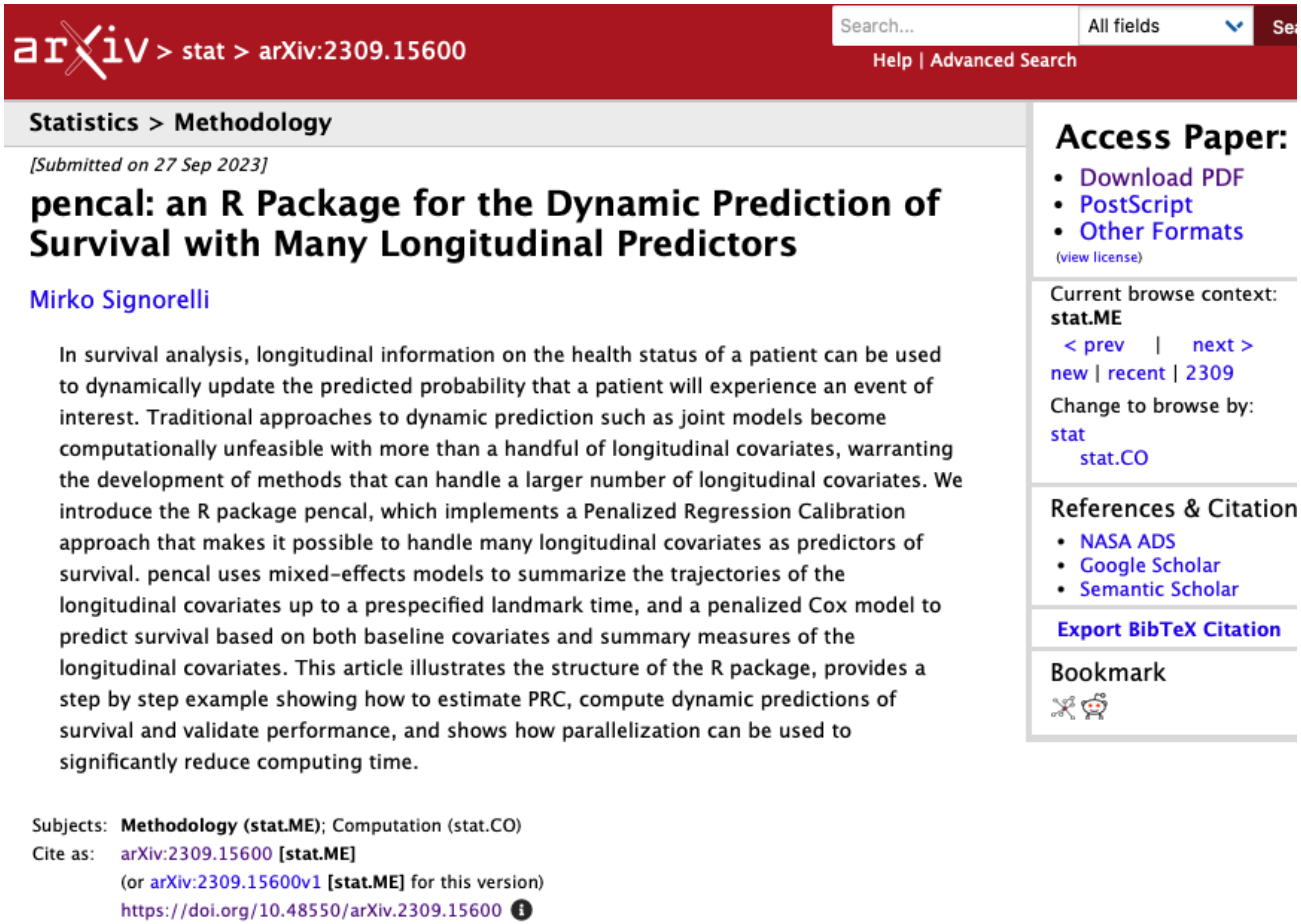
Package overview table

Table 1: Overview of the pencal functions that implement the different modelling steps for the PRC LMM and PRC MLPMM approaches.

Task	PRC LMM	PRC MLPMM
Step 1: estimate the mixed-effects models	`fit_lmms`	`fit_mlpmms`
Step 2: compute the predicted random effects	`summarize_lmms`	`summarize_mlpmms`
Step 3: estimate the penalized Cox model	`fit_prclmm`	`fit_prcmlpmm`
Computation of predicted survival probabilities	`survpred_prclmm`	`survpred_prcmlpmm`
Evaluation of predictive performance	`performance_prc`	`performance_prc`

More about pencal

- Vignette (Signorelli, 2023) available at arXiv:2309.15600:



arXiv > stat > arXiv:2309.15600

Search... All fields Se

Help | Advanced Search

Statistics > Methodology

[Submitted on 27 Sep 2023]

pencal: an R Package for the Dynamic Prediction of Survival with Many Longitudinal Predictors

Mirko Signorelli

In survival analysis, longitudinal information on the health status of a patient can be used to dynamically update the predicted probability that a patient will experience an event of interest. Traditional approaches to dynamic prediction such as joint models become computationally unfeasible with more than a handful of longitudinal covariates, warranting the development of methods that can handle a larger number of longitudinal covariates. We introduce the R package pencal, which implements a Penalized Regression Calibration approach that makes it possible to handle many longitudinal covariates as predictors of survival. pencal uses mixed-effects models to summarize the trajectories of the longitudinal covariates up to a prespecified landmark time, and a penalized Cox model to predict survival based on both baseline covariates and summary measures of the longitudinal covariates. This article illustrates the structure of the R package, provides a step by step example showing how to estimate PRC, compute dynamic predictions of survival and validate performance, and shows how parallelization can be used to significantly reduce computing time.

Subjects: **Methodology (stat.ME)**; Computation (stat.CO)
Cite as: arXiv:2309.15600 **[stat.ME]**
(or arXiv:2309.15600v1 **[stat.ME]** for this version)
https://doi.org/10.48550/arXiv.2309.15600

Access Paper:
- Download PDF
- PostScript
- Other Formats

(view license)

Current browse context: **stat.ME**
< prev | next >
new | recent | 2309
Change to browse by:
stat
stat.CO

References & Citation
- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

Future work

- GLMMs
- Competing risks
- Interval censoring

Strict vs relaxed landmarking with two-step methods

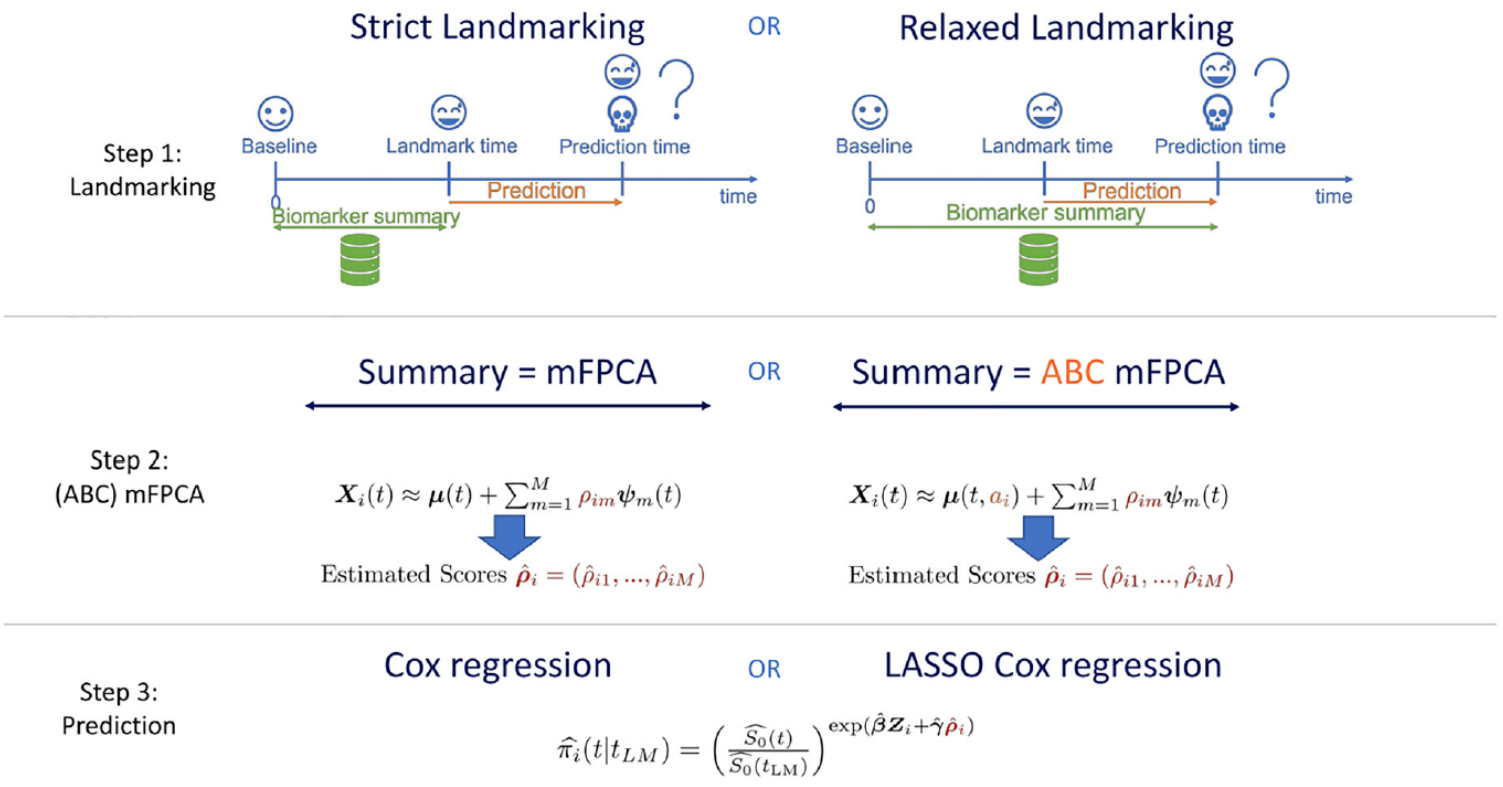
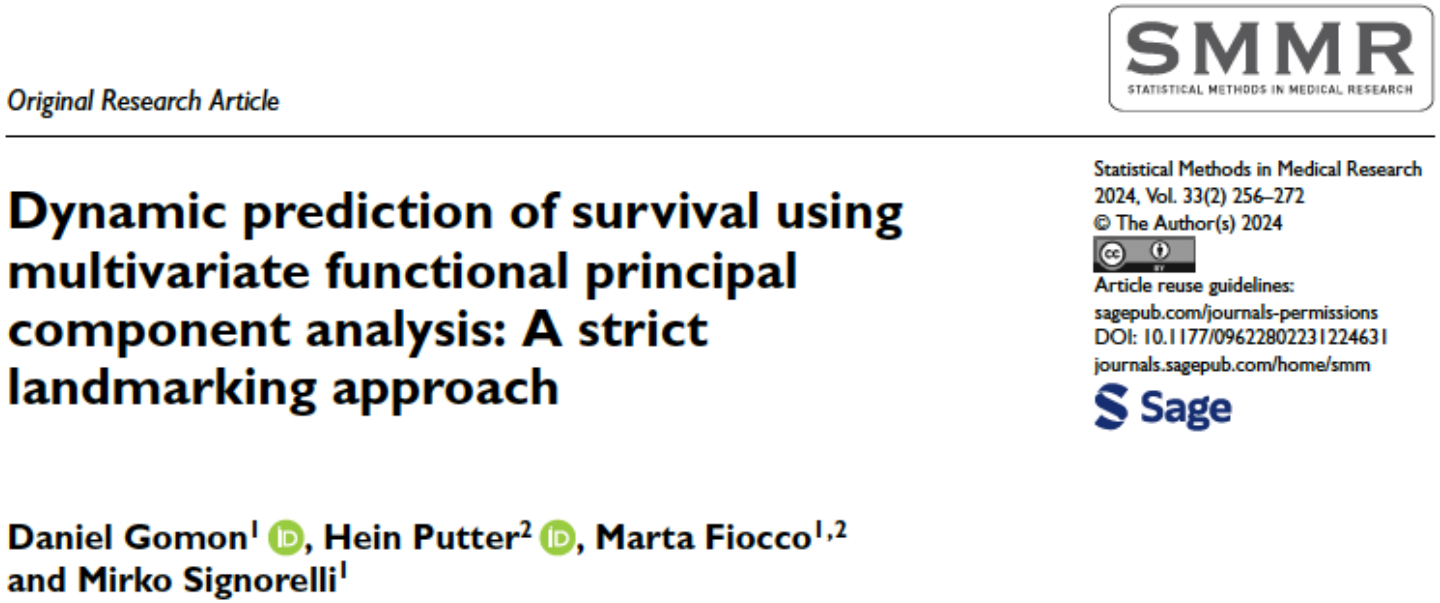


Figure 1. Graphical summary of the methods proposed in Section 2. See also Section 2.6.

Strict vs relaxed landmarking with two-step methods



Original Research Article

SMMR
STATISTICAL METHODS IN MEDICAL RESEARCH

Dynamic prediction of survival using multivariate functional principal component analysis: A strict landmarking approach

Statistical Methods in Medical Research
2024, Vol. 33(2) 256–272

DOI: 10.1177/09622802231224631
journals.sagepub.com/home/smm
Sage

Daniel Gomon[1], Hein Putter[2], Marta Fiocco[1,2] and Mirko Signorelli[1]

References I

Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*(2), 187–202.

Devaux, A., Proust-Lima, C., & Genuer, R. (2023). *Random Forests for time-fixed and time-dependent predictors: The DynForest R package*. arXiv:2302.02670.

Happ, C., & Greven, S. (2018). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, *113*(522), 649–659.

Li, K., & Luo, S. (2019). Dynamic prediction of Alzheimer's disease progression using features of multiple longitudinal outcomes and time-to-event data. *Statistics in Medicine*, *38*(24), 4804–4818.

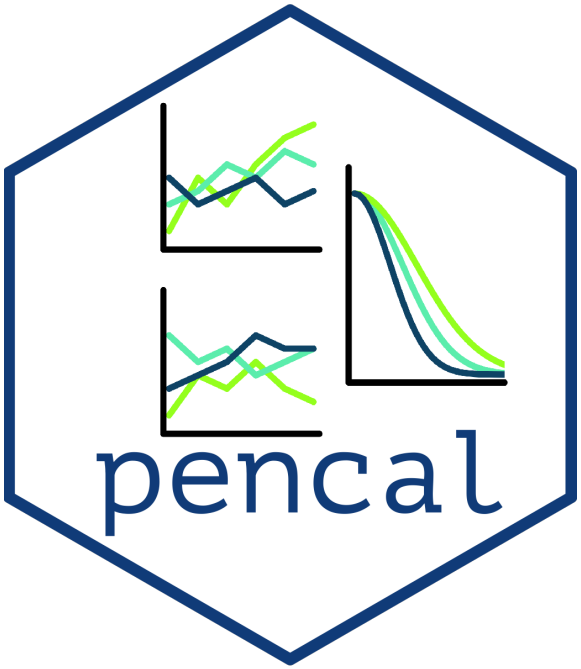
Lin, J., Li, K., & Luo, S. (2021). Functional survival forests for multivariate longitudinal outcomes: Dynamic prediction of Alzheimer's disease progression. *Statistical Methods in Medical Research*, *30*(1), 99–111.

Signorelli, M. (2023). *Pencal: An R Package for the Dynamic Prediction of Survival with Many Longitudinal Predictors*. arXiv:2309.15600.

References II

Signorelli, M., & Retif, S. (2024). *An empirical appraisal of methods for the dynamic prediction of survival with numerous longitudinal predictors*. arXiv:2403.14336.

Signorelli, M., Spitali, P., Szigyarto, C. A., The MARK-MD Consortium, & Tsonaka, R. (2021). Penalized regression calibration: A method for the prediction of survival outcomes using complex longitudinal and high-dimensional data. *Statistics in Medicine*, *40*(27), 6178–6196.

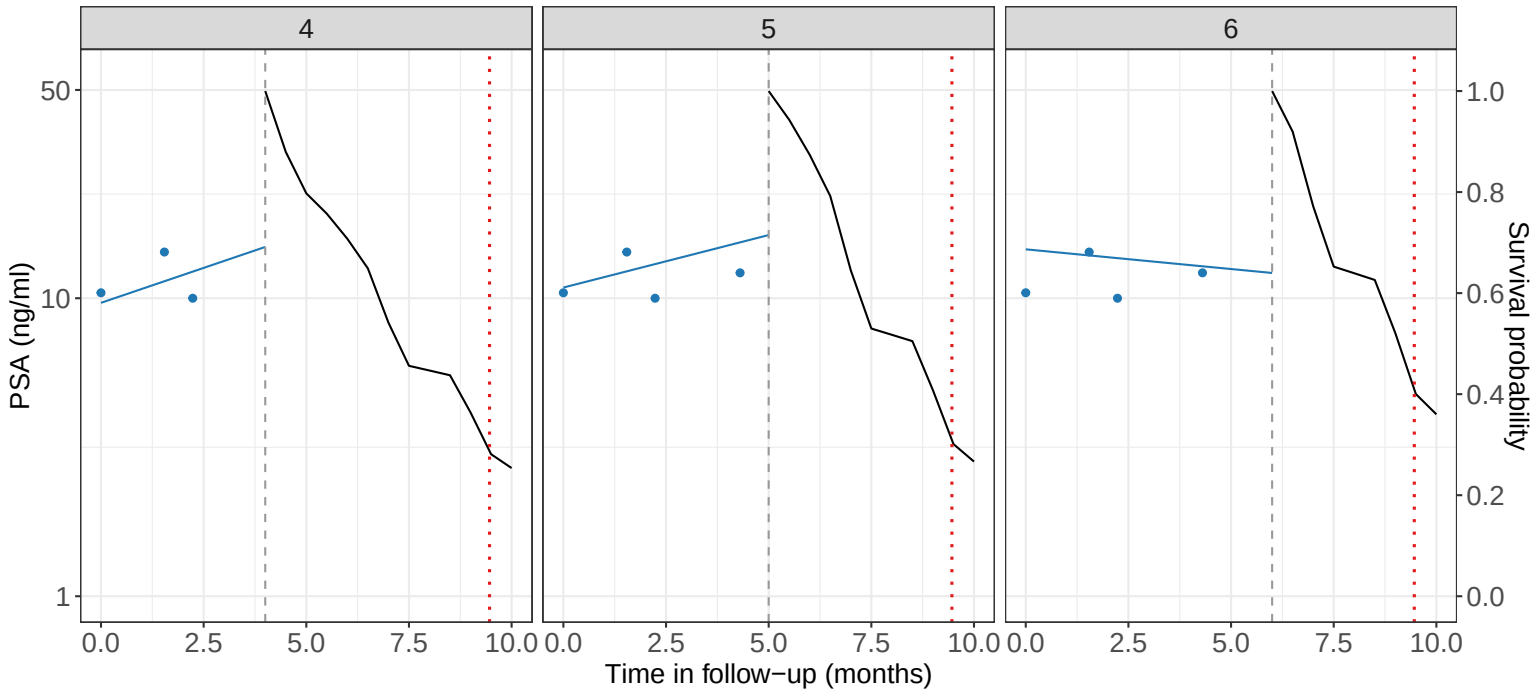


Preprints:

- `pencal` vignette:
  arXiv:2309.15600
- benchmarking study:
  arXiv:2403.14336

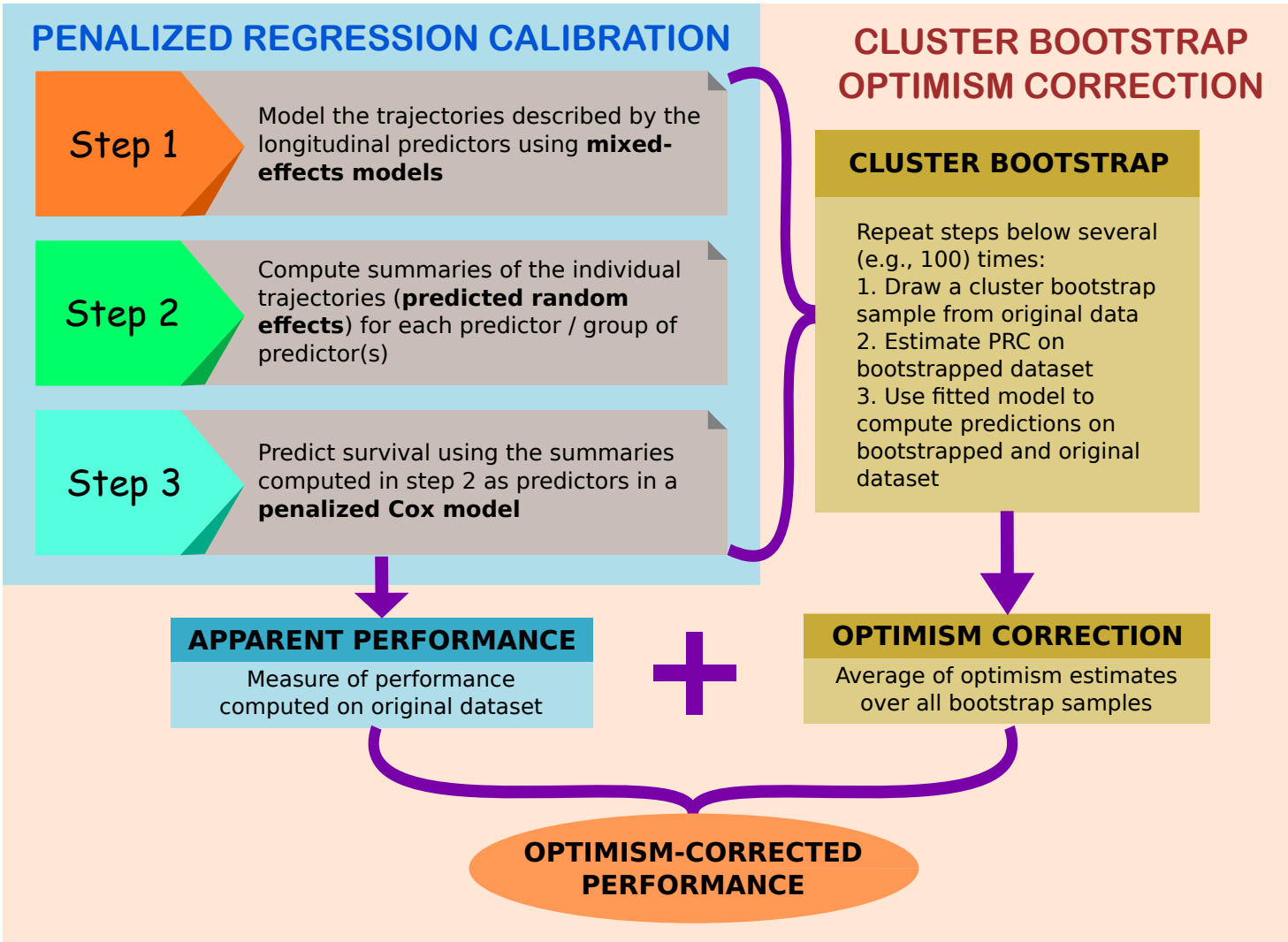
: mirkosignorelli.github.io
: m.signorelli@math.leidenuniv.nl
: @signormirko

Dynamic prediction example



→ go back

Cluster-bootstrap optimism correction



→ go back

Internal validation

- Performance measures: time-dependent AUC, C index, Brier score
- Internal validation of predictive performance:
  - cluster bootstrap optimism correction procedure (Signorelli et al. (2021)) → appendix
  - repeated cross-validation also possible as an alternative

Computing the CBOCP

- To compute the cluster bootstrap optimism correction procedure, rerun steps 1, 2 and 3 specifying `nboots` $= B > 0$ inside `fit_lmms`:

```
step1b = fit_lmms(y.names = long_covs,
                  fixefs = ~ age, ranefs = ~ age | id,
                  long.data = ldata, surv.data = sdata,
                  t.from.base = fuptime,
                  n.boots = 50, n.cores = 8)
step2b = summarize_lmms(step1b, n.cores = 2)
step3b = fit_prclmm(step2b, surv.data = sdata,
                  baseline.covs = ~ baselineAge + sex + treatment,
                  penalty = 'ridge', standardize = T, n.cores = 8)
```

- NB: `n.boots` needs to be specified just in step 1, but it is used also in steps 2 and 3
- `n.cores` allows you to parallelize computations within each step!

Computing the performance measures

```
predPerf = performance_prc(step2 = step2b, step3 = step3b,
                           metric = c('tdauc', 'brier'), times = 3:5,
                           n.cores = 8)
```

Predictive performance

```
predPerf
```

```
## $call
## performance_prc(step2 = step2b, step3 = step3b, metric = c("tdauc",
##     "brier"), times = 3:5, n.cores = 8)
##
## $tdAUC
##   pred.time tdAUC.naive optimism.correction tdAUC.adjusted
## 1         3      0.9439              -0.0056         0.9383
## 2         4      0.9351              -0.0143         0.9208
## 3         5      0.9266              -0.0125         0.9141
##
## $Brier
##   pred.time Brier.naive optimism.correction Brier.adjusted
## 1         3      0.0571               0.0142         0.0713
## 2         4      0.0699               0.0266         0.0965
## 3         5      0.0844               0.0324         0.1168
```