pencal: an R package for the dynamic prediction of survival with many longitudinal predictors

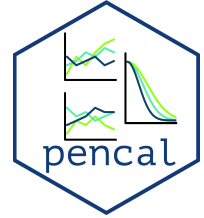
Mirko Signorelli
: mirkosignorelli.github.io
: @signormirko

Mathematical Institute
Leiden University

December 18, 2023 - CMStatistics 2023

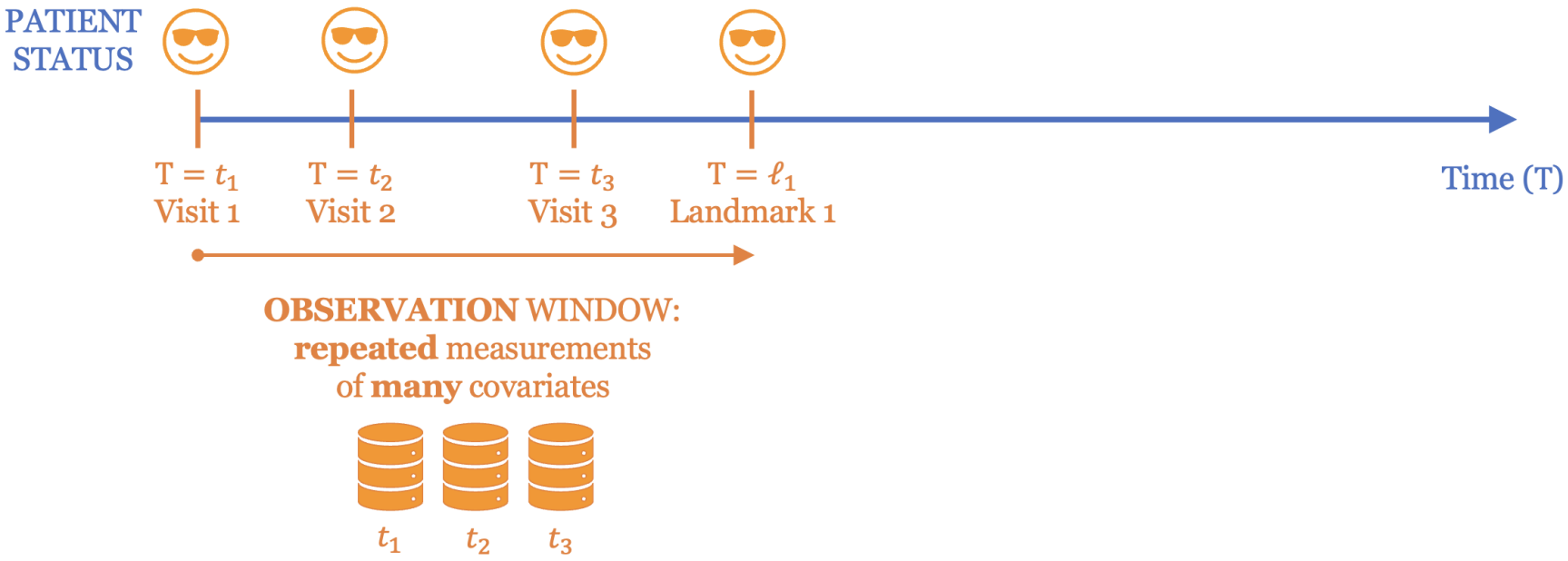
Universiteit Leiden

Vignette: bit.ly/pencal-CMS : @signormirko



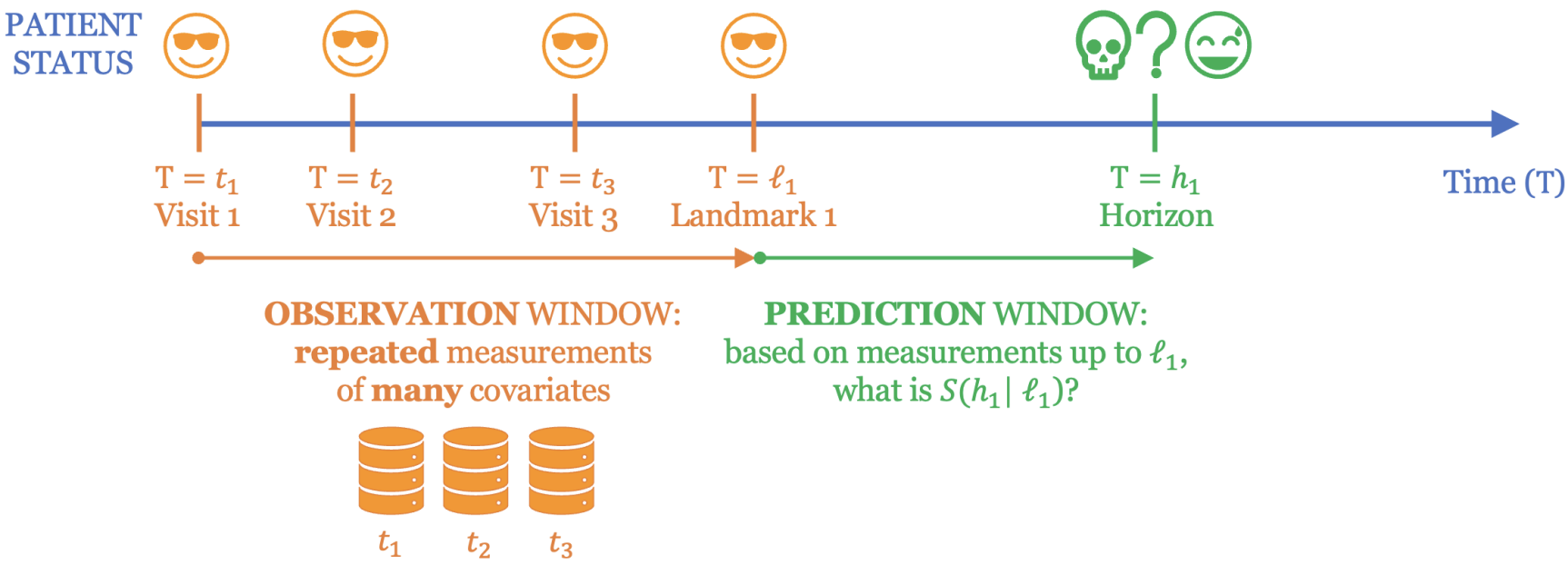
Vignette: bit.ly/pencal-CMS : @signormirko

Dynamic prediction of survival



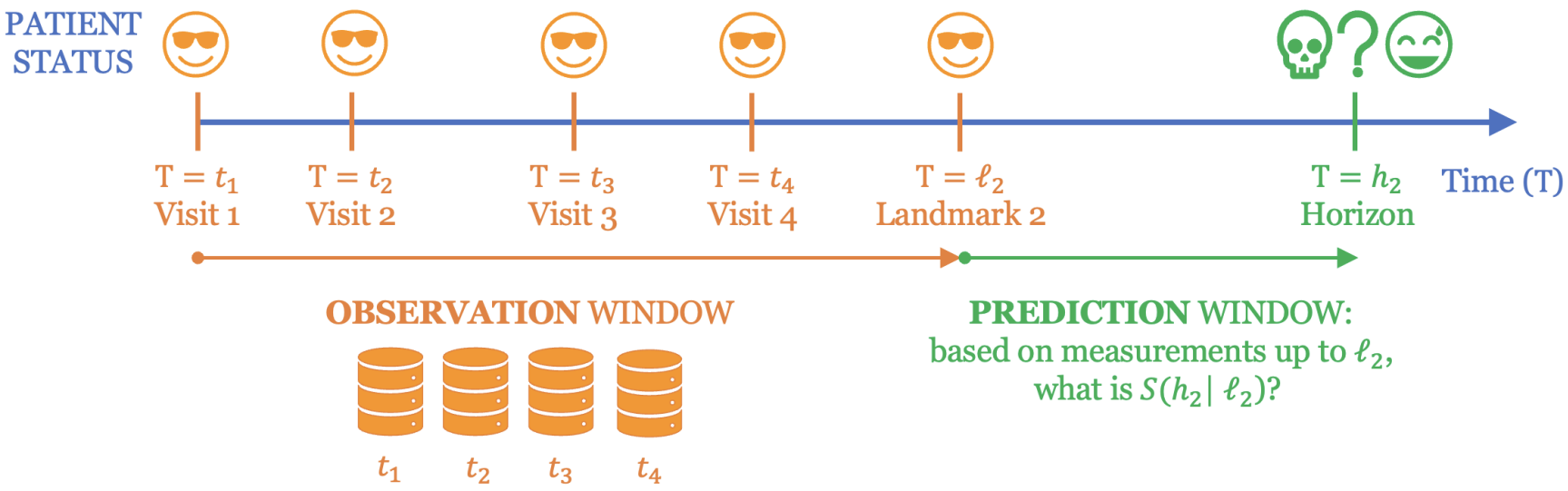
Vignette: bit.ly/pencal-CMS : @signormirko

Dynamic prediction of survival



Vignette: bit.ly/pencal-CMS : @signormirko

Dynamic prediction of survival



Vignette: bit.ly/pencal-CMS : @signormirko

Goal of dynamic prediction

- Goals of dynamic prediction:
  1. predict future survival $S(t|\ell_1)$ using repeated measurements collected over the observation period $[0, \ell_1]$
  2. dynamically update predictions once more information becomes available, i.e. predict $S(t|\ell_2)$ given repeated measurements over $[0, \ell_2]$, $\ell_2 > \ell 1$

Example datasets

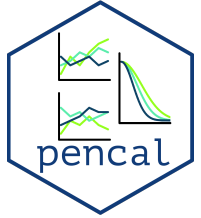
- Two datasets we worked with:

	ROSMAP	Mark-MD
Outcome	Alzheimer's disease diagnosis	Loss of ambulation in Duchenne patients
n	3757	157
Max follow-up	Up to 30 years	Up to 7.4 years
Baseline covariates	5	3
Longitudinal covariates	30	240 antibodies that target 118 proteins

- Datasets can differ substantially wrt n and p

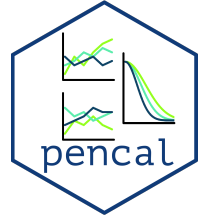
Vignette: bit.ly/pencal-CMS : @signormirko

Methodological 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)
- Methodological problem: how to do dynamic prediction of survival when the predictors are
  1. measured longitudinally (ROSMAP, Mark-MD)
  2. many - potentially high-dimensional setting (ROSMAP, Mark-MD)
  3. and potentially highly-correlated with each other (Mark-MD)

[1] LOCF = Last Observation Carried Forward



The problem: dynamic prediction of survival

The method: Penalized Regression Calibration

The `R` package: `pencal`

Prediction of survival

Evaluation of predictive performance

Package overview

Appendix

Vignette: bit.ly/pencal-CMS : @signormirko

Notation

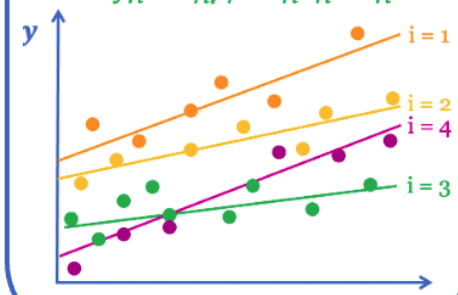
- ℓ = landmark time
- $i \in \{1, ..., n_\ell\}$ subjects who survived up until $t = \ell$
- Observation window - $t \in [0, \ell]$:
  - k baseline covariates x_{qi} measured at $t_i = 0$ (study entry)
  - p longitudinal covariates y_{rij} measured at $t_{i1}, ..., t_{im_i} \in [0, \ell]$
- Prediction window - $t \in [\ell, h]$:
  - T_i^* true survival time
  - C_i censoring time
  - $T_i = \min(T_i^*, C_i)$ observed survival time
  - $\delta_i = I(T_i = T_i^*)$ event indicator

Step 1

For r = 1, ..., p:

LINEAR MIXED MODEL (LMM)

$$y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$$



Step 2

Compute the **predicted random effects**

$$\hat{u}_{ri} = \hat{W}_{ri}(y_{ri} - X_{ri}\hat{\beta}_r)$$

where

$$\hat{W}_{ri} = \hat{D}_r Z_{ri} \hat{V}_{ri}^{-1}$$

Step 3

Estimate

$$h(t|\eta_i) = h_0(t)e^{\eta_i},$$

where

$$\eta_i = x_i^T\tau + \hat{u}_i^T\gamma,$$

using **penalized maximum likelihood** (ridge / lasso / elasticnet)

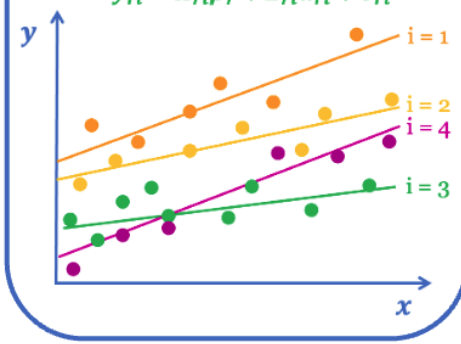
Vignette: bit.ly/pencal-CMS : @signormirko

Step 1

For r = 1, ..., p:

LINEAR MIXED MODEL (LMM)

$$y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$$



Step 2

Compute the **predicted random effects**

$$\hat{u}_{ri} = \hat{W}_{ri}(y_{ri} - X_{ri}\hat{\beta}_r)$$

where

$$\hat{W}_{ri} = \hat{D}_r Z_{ri} \hat{V}_{ri}^{-1}$$

Step 3

Estimate

$$h(t|\eta_i) = h_0(t)e^{\eta_i},$$

where

$$\eta_i = x_i^T\tau + \hat{u}_i^T\gamma,$$

using **penalized maximum likelihood** (ridge / lasso / elasticnet)

Vignette: bit.ly/pencal-CMS : @signormirko

Step 1: model the evolution of the longitudinal covariates

- Step 1: model longitudinal predictors with mixed-effects models
- Two alternatives:
  1. Linear Mixed Models (LMM)
  2. Multivariate Latent Process Mixed Model (MLPMM, Proust-Lima et al. (2013))

LMM approach

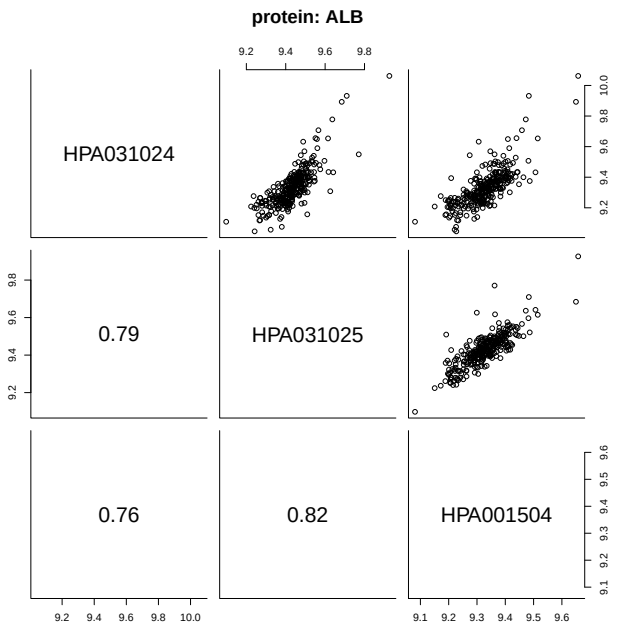
- Fit to each longitudinal Y_r a LMM: $y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$
- Example:

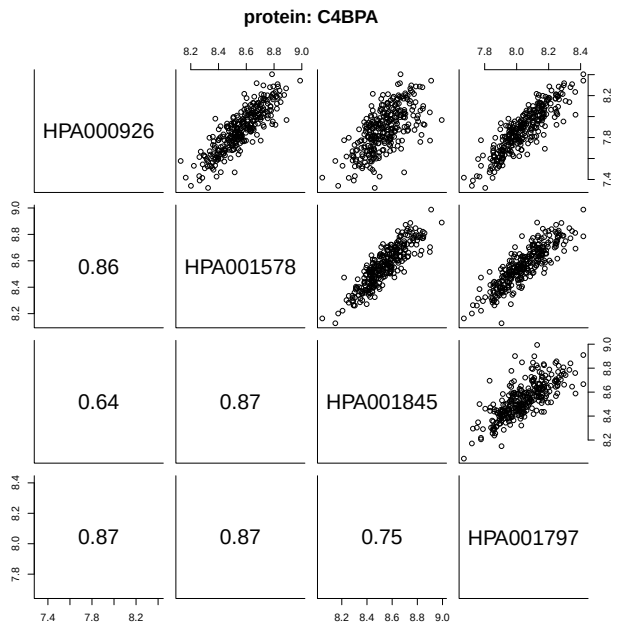
$$y_{rij} = \beta_{r0} + u_{r0i} + (\beta_{r1} + u_{r1i})a_{ij} + \varepsilon_{rij},$$

where $u_{ri} = (u_{r0i}, u_{r1i}) \sim N(0, D_r)$ and $\varepsilon_{ri} \sim N(0, \sigma_r^2 I_{m_i})$

How to handle groups of highly-correlated biomarkers?

- Issue:
  1. LMM approach assumes longitudinal markers to be independent
  2. what if you have groups of highly-correlated biomarkers, like in Signorelli et al. (2020)?





Vignette: bit.ly/pencal-CMS : @signormirko

MLPMM approach

- Suppose that r_s covariates are employed to measure the same underlying phenomenon y_s that cannot be measured directly: $(\mathbf{y_{1s}}, ..., \mathbf{y_{r_s s}})$
- Example: r_s antibodies measured as proxies for protein s
- We can specify a MLPMM for $(\mathbf{y_{1s}}, ..., \mathbf{y_{r_s s}})$ where

$$y_{qsij} = \beta_{qs0} + u_{s0i} + b_{qsi} + (\beta_{qs1} + u_{s1i})a_{ij} + \varepsilon_{qsij} \ \ (\forall q = 1, ..., r_s),$$

with $\varepsilon_{qsij} \sim N_1(0, \sigma^2_{\varepsilon qs})$, and

  - $\mathbf{u}_{si} = (u_{s0i}, u_{s1i}) \sim N_2(0, \Sigma_{us})$: shared random intercept and slope that refer to (latent) underlying quantity ($\rightarrow$ protein)
  - $b_{qsi} \sim N_1(0, \sigma^2_{bqs})$ covariate-specific random intercepts ($\rightarrow$ antibodies)
- Latent variable interpretation (reconstruct latent protein info from measurable antibodies variables)

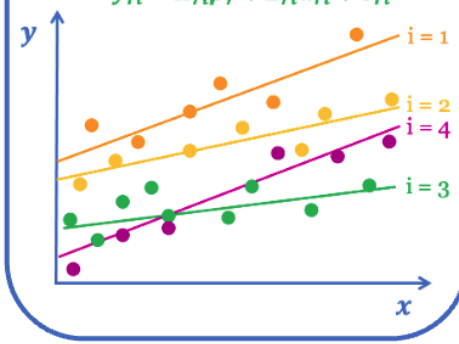
Vignette: bit.ly/pencal-CMS : @signormirko

Step 1

For r = 1, ..., p:

LINEAR MIXED MODEL (LMM)

$$y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$$



Step 2

Compute the **predicted random effects**

$$\hat{u}_{ri} = \hat{W}_{ri}(y_{ri} - X_{ri}\hat{\beta}_r)$$

where

$$\hat{W}_{ri} = \hat{D}_r Z_{ri} \hat{V}_{ri}^{-1}$$

Step 3

Estimate

$$h(t|\eta_i) = h_0(t)e^{\eta_i},$$

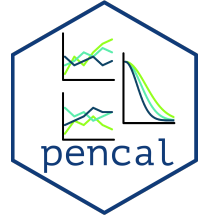
where

$$\eta_i = x_i^T\tau + \hat{u}_i^T\gamma,$$

using **penalized maximum likelihood** (ridge / lasso / elasticnet)

Vignette: bit.ly/pencal-CMS : @signormirko

Step 2: computing the predicted random effects



- Derive subject-specific summaries of the longitudinal trajectories from the mixed-effects models
  - random intercepts $\approx$ different starting levels across subjects
  - random slopes $\approx$ different progression rates between subjects

Step 2: computing the predicted random effects

- Derive subject-specific summaries of the longitudinal trajectories from the mixed-effects models
  - random intercepts $\approx$ different starting levels across subjects
  - random slopes $\approx$ different progression rates between subjects
- For the LMM:

$$\hat{u}_{ri} = E(u_{ri}|Y_{ri} = y_{ri}) = \hat{D}_r Z_i^T \hat{V}_{ri}^{-1}(y_{ri} - X_i\hat{\beta}_r),$$

where $V_{ri} = Z_i D_r Z_i^T + \sigma_r^2 I_{m_i}$ is the marginal covariance matrix of subject i

Step 2: computing the predicted random effects

- For the MLPMM:

$$\left(\hat{u}_{si}, \hat{b}_{si}\right) = E\left(u_{si}, b_{si} | Y_{si} = y_{si}\right) = \begin{bmatrix} Z_i \Sigma_{u_s} \\ \Sigma_{b_s} I_{r_s} \otimes \mathbb{1}_{m_i,1} \end{bmatrix} \Sigma_{y_{si}}^{-1} \dot{y}_{si},$$

where $y_{si} = (y_{1si1}, \ldots, y_{1sim_i}, \ldots, y_{r_s si1}, \ldots, y_{r_s sim_i})^T$, $\dot{y}_{si}$ is the equivalent of y_{si} with $\dot{y}_{qsij} = y_{qsij} - \beta_{qs0} - \beta_{qs1} a_{ij}$ as entries, Z_i is the random-effects design matrix associated to y_{si}, $\Sigma_{b_s} = \begin{bmatrix} \sigma^2_{b1s} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma^2_{br_s s} \end{bmatrix}$,

$\Sigma_{\varepsilon_s} = \begin{bmatrix} \sigma^2_{\varepsilon 1s} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma^2_{\varepsilon r_s s} \end{bmatrix}$, $\Sigma_{u_s} = \begin{bmatrix} \sigma^2_{us0} & \sigma_{us0,us1} \\ \sigma_{us0,us1} & \sigma^2_{us1} \end{bmatrix}$ and

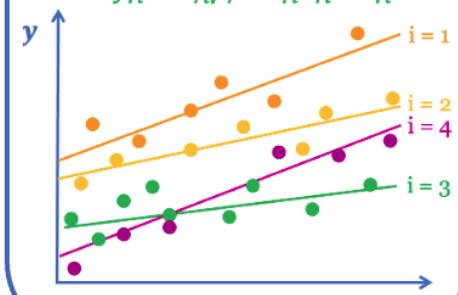
$\Sigma_{y_{si}} = Z_i \Sigma_{us} Z_i^T + I_{r_s} \otimes \Sigma_{\varepsilon s} I_{m_i} + I_{r_s} \otimes \Sigma_{bs} \mathbb{1}_{m_i,m_i}$

Step 1

For r = 1, ..., p:

LINEAR MIXED MODEL (LMM)

$$y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$$



Step 2

Compute the **predicted random effects**

$$\hat{u}_{ri} = \hat{W}_{ri}(y_{ri} - X_{ri}\hat{\beta}_r)$$

where

$$\hat{W}_{ri} = \hat{D}_r Z_{ri} \hat{V}_{ri}^{-1}$$

Step 3

Estimate

$$h(t|\eta_i) = h_0(t)e^{\eta_i},$$

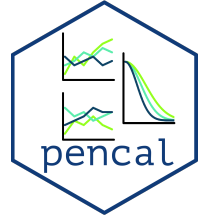
where

$$\eta_i = x_i^T\tau + \hat{u}_i^T\gamma,$$

using **penalized maximum likelihood** (ridge / lasso / elasticnet)

Vignette: bit.ly/pencal-CMS : @signormirko

Step 3: prediction of survival



- Cox model linking survival outcome to baseline covariates and summaries of longitudinal covariates:

$$h(t_i|x_i, \hat{u}_{0i}, \hat{u}_{1i}) = h_0(t_i)\exp(\eta_i), \tag{1}$$

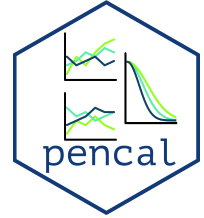
$$\eta_i = \sum_{q=1}^{k} \theta_q x_{qi} + \sum_{r=1}^{p} \gamma_r \hat{u}_{r0i} + \sum_{r=1}^{p} \delta_r \hat{u}_{r1i}$$

- (θ, γ, δ) large, potentially high-dimensional $\Rightarrow$ we estimate it using penalized maximum likelihood

$$\max_{\xi,\gamma,\delta} \ \ell(\xi, \gamma, \delta) - \lambda p(\xi, \gamma, \delta; \alpha)$$

- Penalty functions: ridge (ℓ^2, recommended), elastic net, lasso (ℓ^1)
- Predicted survival: $\hat{S}(h|\ell) = \hat{S}(h) = e^{-\int_0^h \hat{h}_0(s)e^{\hat{\eta}_i} ds}$

Vignette: bit.ly/pencal-CMS : @signormirko



The problem: dynamic prediction of survival

The method: Penalized Regression Calibration

The `R` package: `pencal`

Prediction of survival

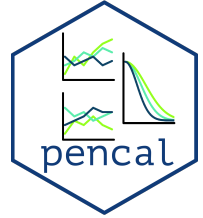
Evaluation of predictive performance

Package overview

Appendix

Vignette: bit.ly/pencal-CMS : @signormirko

Where to find the package



- Method 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

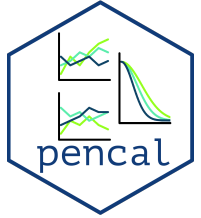
Vignette: bit.ly/pencal-CMS : @signormirko

Example dataset

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

- Data from the PBC2 clinical trial (1974-1984)
  - n = 312, k = 3, p = 7
  - Outcome: time to death
  - Follow-up up to 14.3 years

Data preparation



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

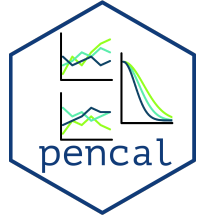
```
# remove subjects with event / censoring before landmark
lmark = 2
sdata = subset(sdata, time > lmark)
ldata = subset(ldata, id %in% sdata$id)

# remove repeated measurements taken after landmark
ldata = subset(ldata, fuptime <= 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
```

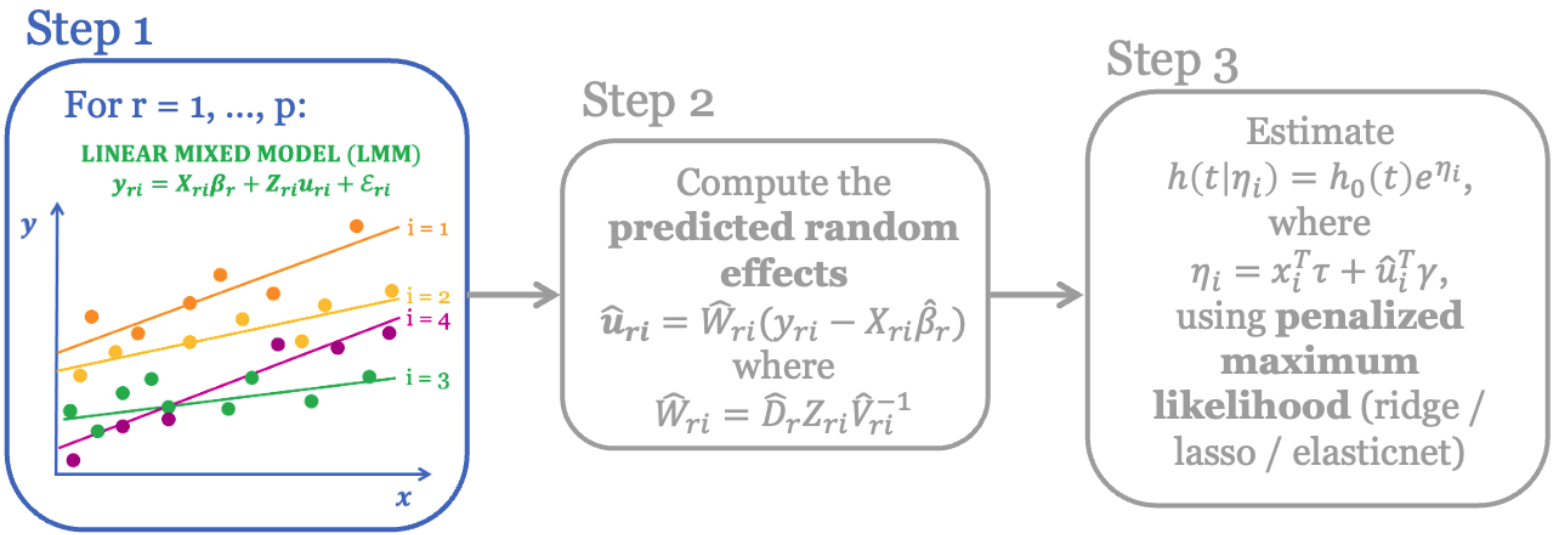
Vignette: bit.ly/pencal-CMS : @signormirko

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



Vignette: bit.ly/pencal-CMS : @signormirko

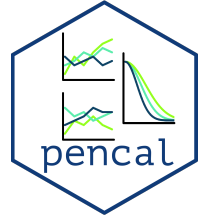
Step 1: estimating the LMMs

```
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)
```

Vignette: bit.ly/pencal-CMS : @signormirko

Extracting output from the fitted LMMs




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

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

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

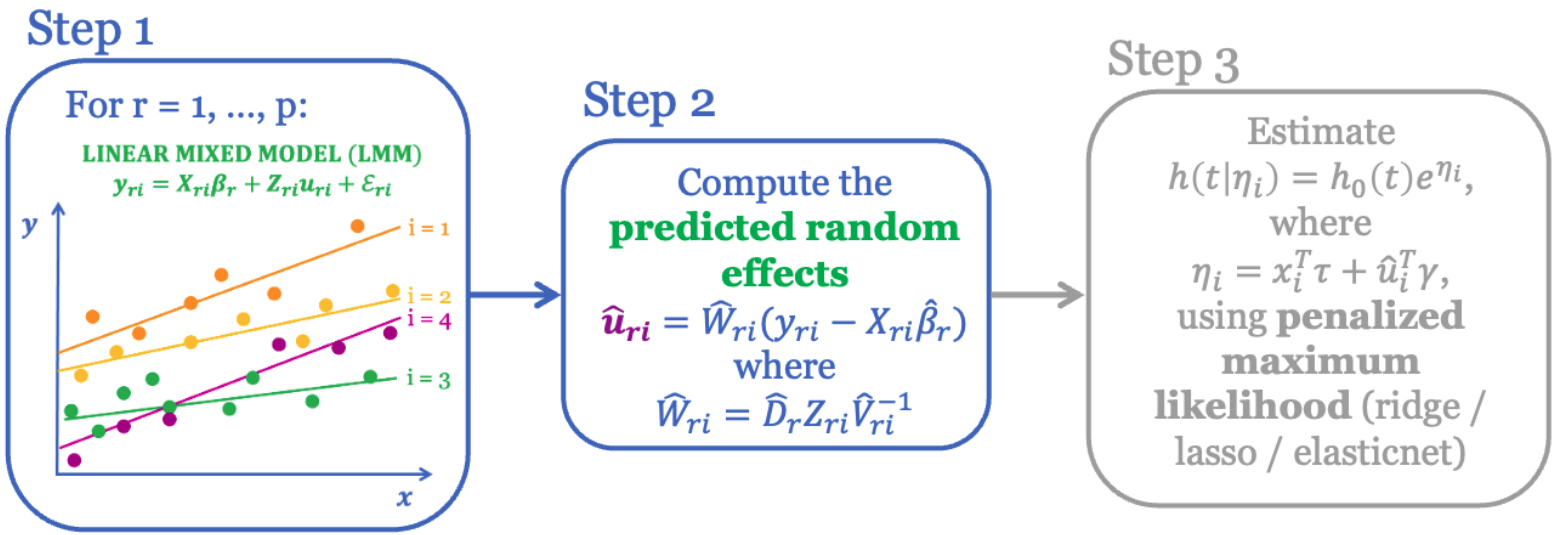
```
##                Value Std.Error  DF t-value p-value
## (Intercept)  0.5183    0.2788 566  1.8590  0.0636
## age         -0.0010    0.0055 566 -0.1884  0.8506
```

```
getlmm(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
```

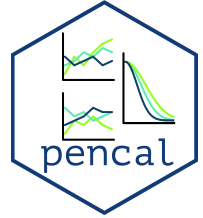
Vignette: bit.ly/pencal-CMS : @signormirko

Step 2: computing the predicted random effects



Vignette: bit.ly/pencal-CMS : @signormirko

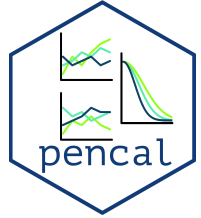
Step 2: computing the predicted random effects




```
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
```

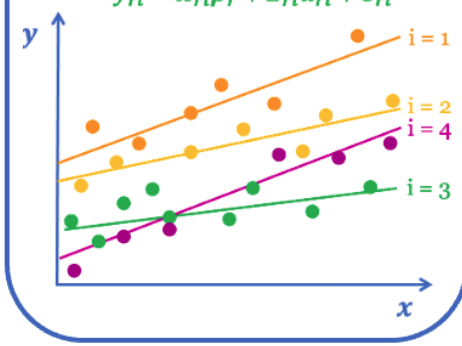
Step 3: estimate the penalized Cox model

Step 1

For r = 1, ..., p:

LINEAR MIXED MODEL (LMM)

$y_{ri} = X_{ri}\beta_r + Z_{ri}u_{ri} + \varepsilon_{ri}$



Step 2

Compute the **predicted random effects**

$\hat{u}_{ri} = \hat{W}_{ri}(y_{ri} - X_{ri}\hat{\beta}_r)$

where

$\hat{W}_{ri} = \hat{D}_r Z_{ri} \hat{V}_{ri}^{-1}$

Step 3

Estimate

$h(t|\eta_i) = h_0(t)e^{\eta_i}$,

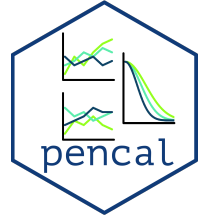
where

$\eta_i = x_i^T\tau + \hat{u}_i^T\gamma$,

using **penalized maximum likelihood** (ridge / lasso / elasticnet)

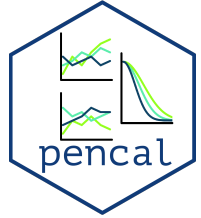
Vignette: bit.ly/pencal-CMS : @signormirko

Step 3: estimate the penalized Cox model




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

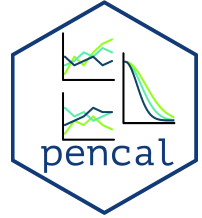
Step 3: fitted model



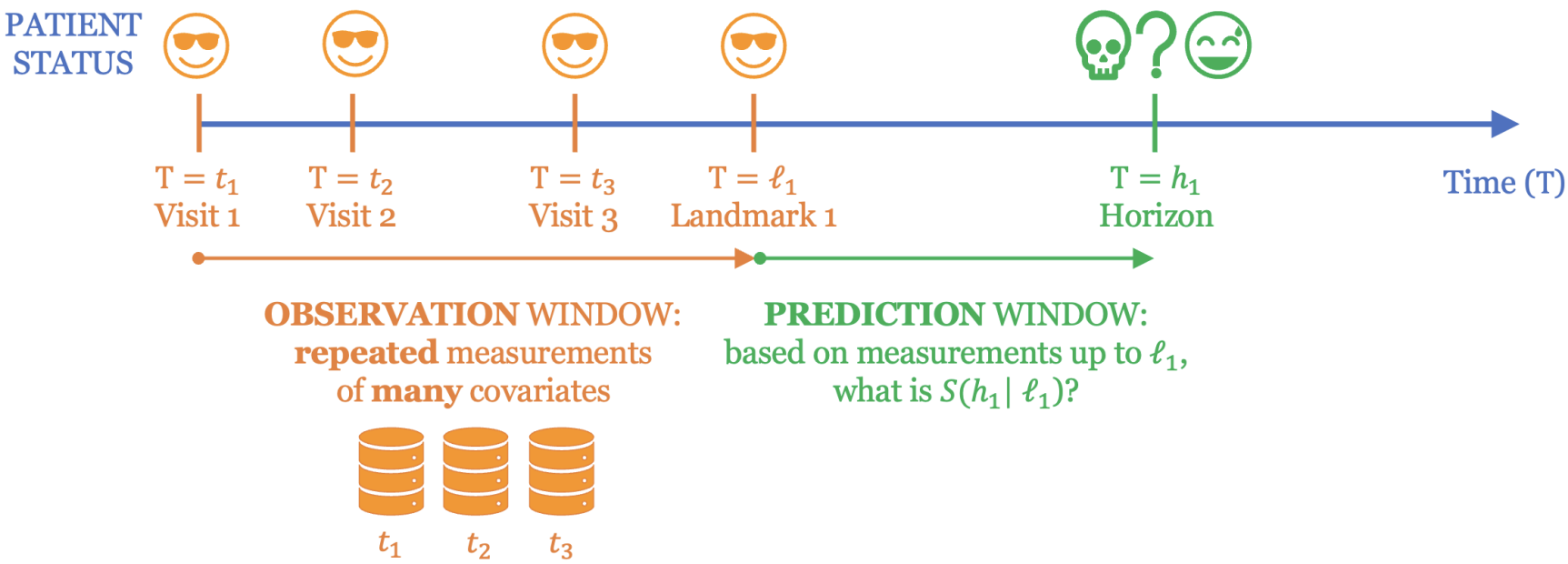

```
print(step3, digits = 3)
```

```
## Fitted model: PRC-LMM
## Penalty function used: ridge
## Sample size: 278
## Number of events: 107
## Bootstrap optimism correction: not computed
## Penalized likelihood estimates (rounded to 3 digits):
##   baselineAge sexfemale treatmentD-penicil logSerBil_b_int
## 1       0.051     -0.31               0.019           0.583
##   logSerBil_b_age logSerChol_b_int logSerChol_b_age
## 1         147.094             0.065           -7.513
##   albumin_b_int albumin_b_age logAlk_b_int logAlk_b_age
## 1         -1.55      34804.81        0.086       -0.217
##   logSGOT_b_int logSGOT_b_age platelets_b_int
## 1         0.164       260.291          -0.001
##   platelets_b_age logProthr_b_int logProthr_b_age
## 1          -0.043           3.053         -92.509
```

Vignette: bit.ly/pencal-CMS : @signormirko



Back to our goal: predicting survival



Vignette: bit.ly/pencal-CMS : @signormirko

Prediction of survival

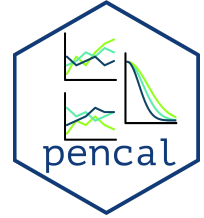
```
Shat = survpred_prclmm(step1, step2, step3, times = 3:5)
```

- 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 0.9602048 0.9211106 0.8788494
## 3  3 0.8667790 0.7487764 0.6346572
## 4  4 0.8232941 0.6747116 0.5388305
## 5  5 0.9443086 0.8905143 0.8334098
```

- Prediction for new subjects? Possible through additional arguments `new.longdata` and `new.basecovs`



Vignette: bit.ly/pencal-CMS : @signormirko

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

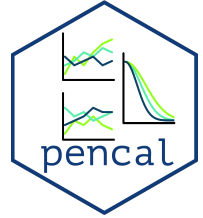
Computing the CBOCP

- 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)
```

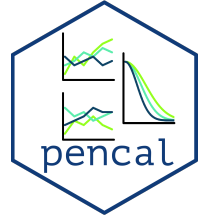
- NB: `n.boots` needs to be specified just in step 1, but it is used also in steps 2 and 3
- `n.cores` allows you to parallelize computations within each step!

Computing the performance measures



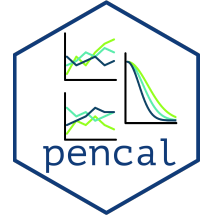

```
predPerf = performance_prc(step2 = step2b, step3 = step3b,
                           metric = c('tdauc', 'brier'), times = 3:5,
                           n.cores = 8)
```

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      0.9434              -0.0064         0.9370
## 2         4      0.9348              -0.0160         0.9188
## 3         5      0.9273              -0.0134         0.9139
##
## $Brier
##   pred.time Brier.naive optimism.correction Brier.adjusted
## 1         3      0.0556               0.0127         0.0683
## 2         4      0.0679               0.0239         0.0918
## 3         5      0.0823               0.0286         0.1109
```



Vignette: bit.ly/pencal-CMS : @signormirko

Package overview table

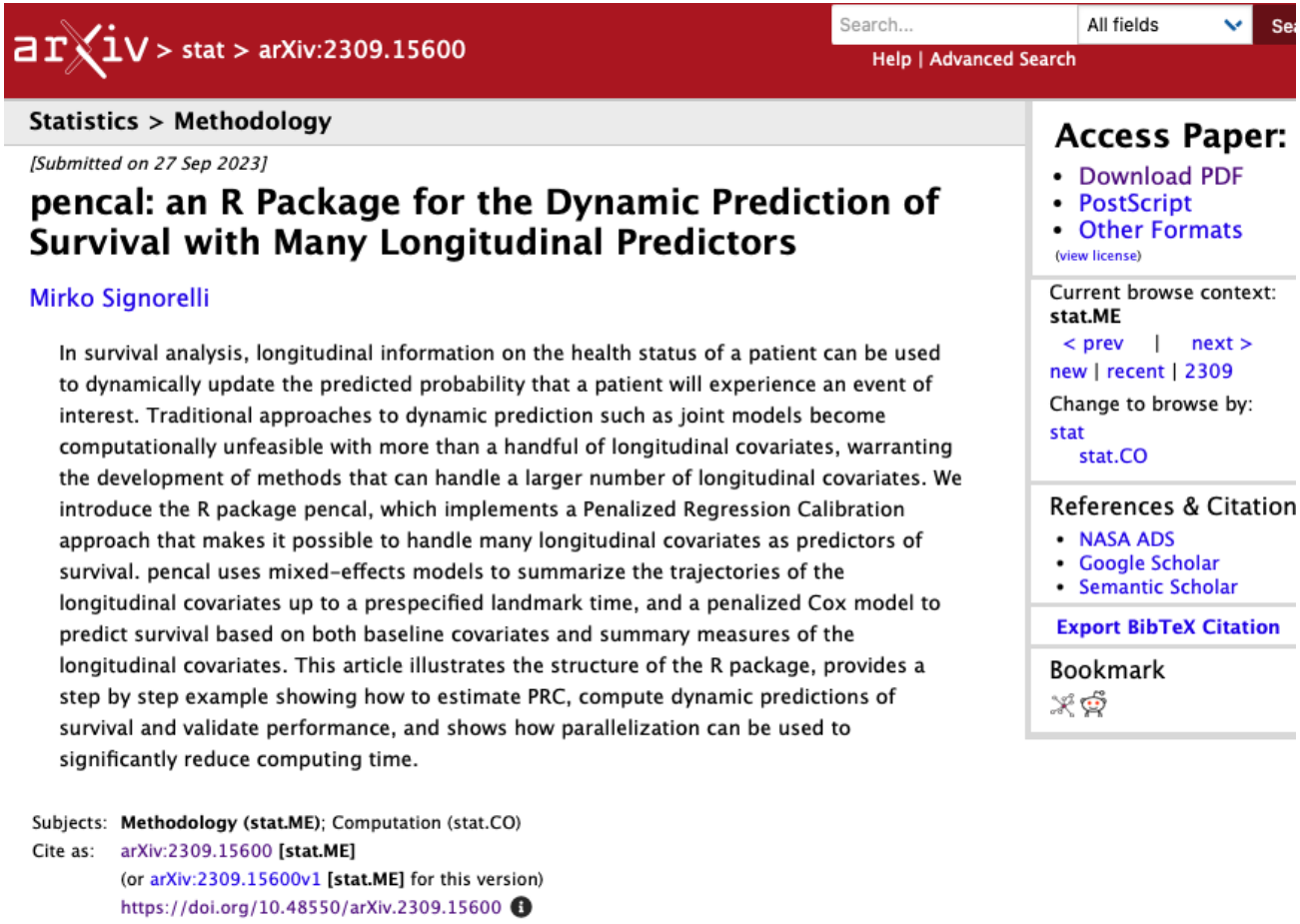
Table 1: Overview of the pencal functions that implement the different modelling steps for the PRC LMM and PRC MLPMM approaches.

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`

Vignette: bit.ly/pencal-CMS : @signormirko

More about pencal

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



arXiv > stat > arXiv:2309.15600

Search... All fields

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

Vignette: bit.ly/pencal-CMS : @signormirko

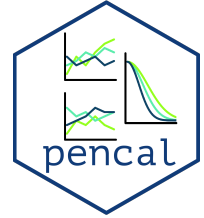
References I

Proust-Lima, C., Amieva, H., & Jacqmin-Gadda, H. (2013). Analysis of multivariate mixed longitudinal data: A flexible latent process approach. *British Journal of Mathematical and Statistical Psychology*, *66*(3), 470–487.

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

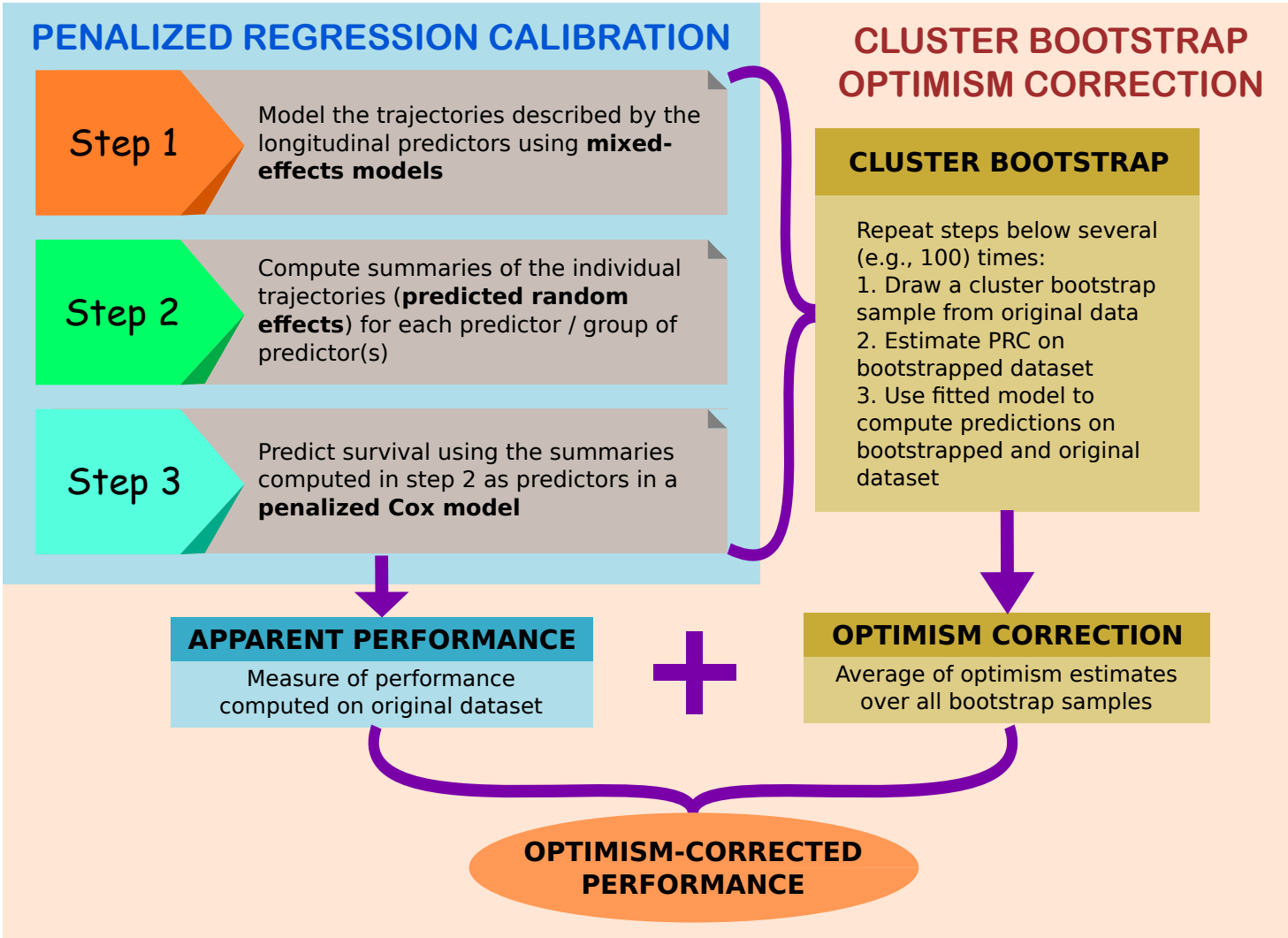
Signorelli, M., Ayoglu, B., Johansson, C., Lochmüller, H., Straub, V., Muntoni, F., Niks, E., Tsonaka, R., Person, A., Aartsma-Rus, A., Nilsson, P., Al-Khalili Szigyarto, C., & Spitali, P. (2020). Longitudinal serum biomarker screening identifies MDH2 as candidate prognostic biomarker for Duchenne muscular dystrophy. *Journal of Cachexia, Sarcopenia and Muscle*, *11*(2), 505–517.

Signorelli, M., Spitali, P., Al-Khalili Sgyziarto, C., 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*.



Vignette: bit.ly/pencal-CMS : @signormirko

Cluster-bootstrap optimism correction



→ go back

Vignette: bit.ly/pencal-CMS : @signormirko