A general framework and implementation for variance modelling in joint model settings

Michael J. Crowther

Biostatistician
Department of Medical Epidemiology and Biostatistics,
Karolinska Institutet, Sweden
and
Consultant Software Developer

ISCB Lyon
21st July 2021

mjcrowther.co.uk michael.crowther@ki.se @Crowther_MJ

My background

- I mainly work in methodology; developing new statistical models and software
- My applied work is generally in cardiovascular disease and cancer
- My main areas of research are:
  - joint modelling of longitudinal and survival data
  - multi-state survival analysis
  - software development
- Some current projects include multiple timescale multi-state models, and interval-censored competing risks

A disclaimer

- I work in Stata... I am currently working as a consultant developer for StataCorp
- All of my work in joint modelling, including the packages I've developed has been done as part of my academic work
- The lecture slides will be biased towards Stata, (the syntax between the Stata and R versions of `merlin` are similar)

Introduction

Motivation

- More data $\rightarrow$ more questions
  - need for appropriate statistical modelling techniques, and implementations

Motivation

- More data $\rightarrow$ more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...

Motivation

- More data $\rightarrow$ more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...
- The standard (yet often still ignored) challenges
  - time-dependent effects, non-linear covariate effects

Motivation

- More data $\rightarrow$ more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...
- The standard (yet often still ignored) challenges
  - time-dependent effects, non-linear covariate effects
- The neglected challenges
  - Multivariate outcomes
  - Within-patient variability
  - Informative observations times

Motivation

- More data $\rightarrow$ more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...
- The standard (yet often still ignored) challenges
  - time-dependent effects, non-linear covariate effects
- The neglected challenges
  - Multivariate outcomes
  - Within-patient variability
  - Informative observations times

We need modelling frameworks that can accommodate a lot of different things

Motivation

What is joint modelling?

- Joint/Simultaneous Modelling

Motivation

What is joint modelling?

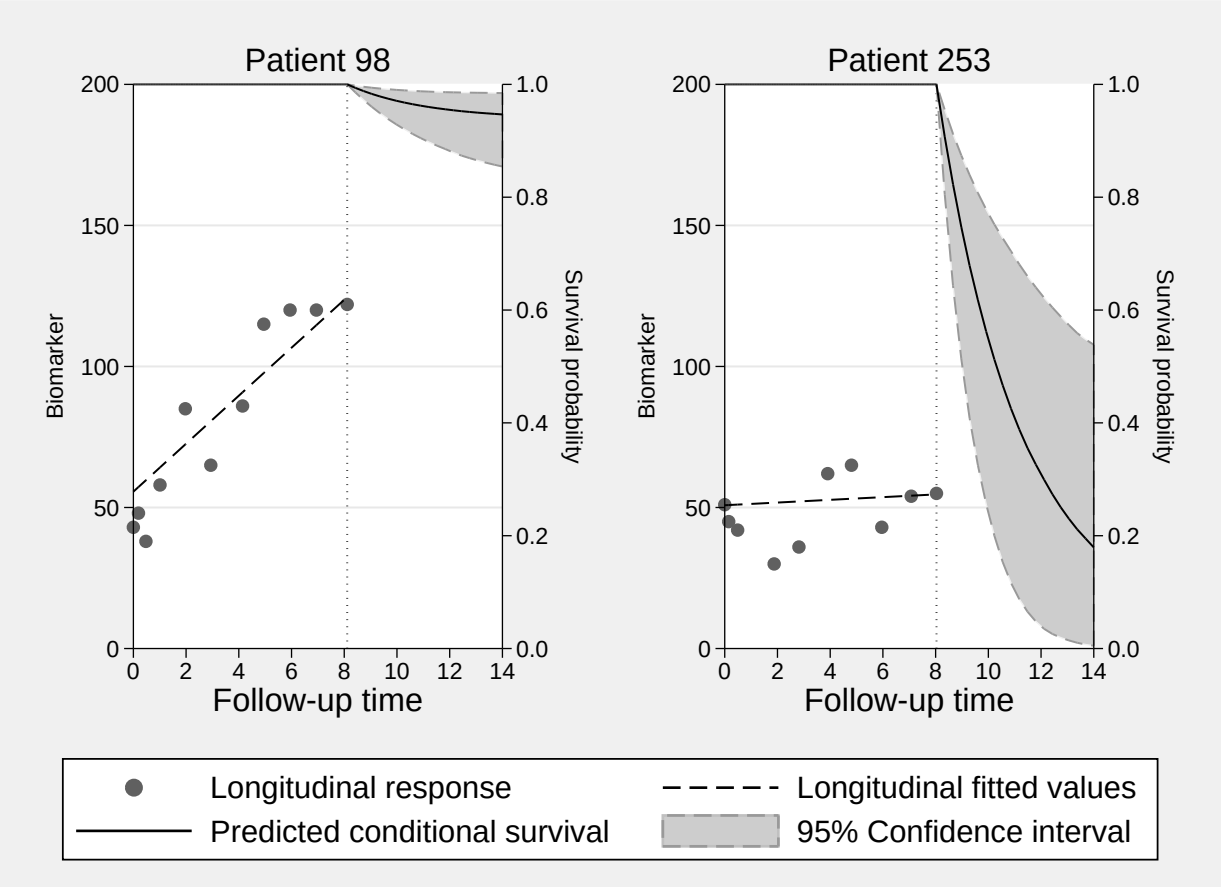
- Joint/Simultaneous Modelling
- 2 broad inter-linked processes:
  - Biomarker process (longitudinal) [mixed] model
  - Time to clinical outcome process (survival/time-to-event) model

Motivation

What is joint modelling?

- Joint/Simultaneous Modelling
- 2 broad inter-linked processes:
  - Biomarker process (longitudinal) [mixed] model
  - Time to clinical outcome process (survival/time-to-event) model
- Focus may be...
  - Estimating biomarker profile/trajectory allowing for informative dropout, e.g. death
  - Estimating relationship between underlying [adjusting for measurement error] biomarker profile/trajectory and clinical outcome

Joint longitudinal-survival models



Linking via - current value, gradient, AUC, random effects...

Joint modelling of longitudinal and survival data

Setting the scene

Modelling a continuous outcome, such as blood pressure, over time, using a linear mixed effects model

$$y_i(t) = m_i(t) + e_i(t), \qquad e_i(t) \sim N(0, \sigma^2)$$
$$m_i(t) = \boldsymbol{X_i}^T(t)\boldsymbol{\beta} + \boldsymbol{Z_i}^T(t)\boldsymbol{b_i}, \qquad \boldsymbol{b_i} \sim N(0, \Sigma)$$

Setting the scene

Modelling a continuous outcome, such as blood pressure, over time, using a linear mixed effects model

$$y_i(t) = m_i(t) + e_i(t), \qquad e_i(t) \sim N(0, \sigma^2)$$
$$m_i(t) = \boldsymbol{X_i}^T(t)\beta + \boldsymbol{Z_i}^T(t)\boldsymbol{b_i}, \qquad \boldsymbol{b_i} \sim N(0, \Sigma)$$

Modelling a time-to-event outcome, such as time to death, using a proportional hazards model, and linking them together

$$h_i(t) = h_0(t)\exp(\phi^T \boldsymbol{v_i} + \alpha m_i(t))$$

Setting the scene

We are interested in what we can do if the longitudinal and survival outcomes are related? This gives rise to such research questions as:

Setting the scene

We are interested in what we can do if the longitudinal and survival outcomes are related? This gives rise to such research questions as:

- What if the trajectory of blood pressure, i.e. how it changes over time, impacts the risk of death?
- If patients with higher blood pressure are more likely to die, will this affect our estimates of the trajectory of BP over time?

Motivation

Joint longitudinal-survival models - extensions

- Competing risks (Li et al., 2009)
- Different types of outcomes (Rizopoulos et al., 2008)
- Multiple continuous outcomes (Lin et al., 2002)
- Delayed entry (Crowther et al., 2016)
- Recurrent events and a terminal event (Król et al., 2016)
- Prediction (Barrett and Su, 2017)
- Many others...

Motivation

Joint longitudinal-survival models - software

- `stjm` in Stata (Crowther et al., 2013)
- `gsem` in Stata
- `frailtypack` in R (Król et al., 2017)
- `joineR` & `joineRML` in R (Philipson et al., 2012)
- `JM` and `JMBayes` in R (Rizopoulos, 2010, 2015)
- Many others...

The goal

A unified framework for data analysis and methods development
(Crowther, 2020)

- Multiple outcomes of varying types
- Measurement schedule can vary across outcomes
- Any number of levels and random effects
- Sharing and linking random effects between outcomes
- Sharing functions of the expected value of other outcomes
- A reliable estimation engine
- Easily extendable by the user
- ...

A general level likelihood

Exploiting conditional independence amongst level $l-1$ units, given the random effects at higher levels,

$$ll(\beta) = \log \int \phi(b^{(L)}|\Sigma^{(L)}) \prod p^{(L-1)}(y|x, b^L, \beta) \, \mathrm{d}b^{(L)}$$

where, for $l = 2, \ldots, L$

$$p^{(l)}(y|x, B^{l+1}, \beta) = \int \phi(b^{(l)}|\Sigma^{(l)}) \prod p^{(l-1)}(y|x, B^l, \beta) \, \mathrm{d}b^{(l)}$$

Extended linear predictor

$$\eta_i = g_i(E[y_i|X, b]) = \sum_{r=1}^{R_i} \prod_{s=1}^{S_{ir}} \psi_{irs}$$

where $g_i()$ is the link function for the ith outcome. To maintain generality, $\psi_{irs}(t)$ can take many forms, including,

$$\begin{aligned}
&\psi_{irs}(t) = X \\
&\psi_{irs}(t) = \beta \\
&\psi_{irs}(t) = b \\
&\psi_{irs}(t) = q(t) \\
&\psi_{irs}(t) = d_{rs}(E[y_j]), \qquad \text{where } j = 1, \ldots, k, j \neq i
\end{aligned}$$

Extended linear predictor

$$\eta_i = g_i(E[y_i|X, b]) = \sum_{r=1}^{R_i} \prod_{s=1}^{S_{ir}} \psi_{irs}$$

where $g_i()$ is the link function for the ith outcome. To maintain generality, $\psi_{irs}(t)$ can take many forms, including,

$$\begin{aligned}
&\psi_{irs}(t) = X \\
&\psi_{irs}(t) = \beta \\
&\psi_{irs}(t) = b \\
&\psi_{irs}(t) = q(t) \\
&\psi_{irs}(t) = d_{rs}(E[y_j]), \qquad \text{where } j = 1, \ldots, k, j \neq i
\end{aligned}$$

Ancillary parameters could also have their own complex predictor

merlin

Title

merlin — Mixed effects regression for linear and non-linear models

Syntax

merlin *models* [*if*] [*in*] [, *options*]

where *models* are the model specifications; see **merlin models**.

options	Description
model_description_options	fully define, along with *models*, the model to be fit
estimation_options	method used to obtain estimation results, including specifying initial values
reporting_options	reporting of estimation results

Also see **merlin postestimation** for features available after estimation.

merlin

family	Description
gaussian	Gaussian (normal)
bernoulli	Bernoulli
beta	beta
poisson	Poisson
nbinomial	negative binomial with mean dispersion
ologit	ordinal response with logit link
oprobit	ordinal response with probit link
gamma	gamma distribution
lquantile [, quantile(#)]	linear quantile model with asymmetric Laplace distribution
exponential [, *survival*`]`	exponential
gompertz [, *survival*`]`	Gompertz
rp [, *survival rpopts*`]`	Royston–Parmar model on the log cumulative hazard scale
loghazard [, *survival*`]`	general log hazard model
logchazard [, *survival*`]`	general log cumulative hazard model
weibull [, *survival*`]`	Weibull
user [, *user survival*`]`	user-defined
null	does not contribute to the log-likelihood; see details

survival	Description
failure(*varname***)**	indicator for failure event
linterval(*varname***)**	lower interval time for interval censored observations
ltruncated(*varname***)**	entry time for left-truncated/delayed-entry model
bhazard(*varname***)**	expected mortality rate at event times, invokes a relative survival model

merlin

element	Description
varname	a variable in the dataset
M#[*levelvar...***]**	a random effect at the specified level; see *details*
fp(*varname***,** *fp_opts***)**	a fractional polynomial function of *varname*
rcs(*varname***,** *rcs_opts***)**	a restricted cubic spline function of *varname*
bs(*varname***,** *bs_opts***)**	a *B*-spline function of *varname*
mf(*function_name***)**	a user-defined Mata function; see *details*
EV[*depvar***|#]**	expected value of another outcome; see *details*
dEV[*depvar***|#]**	d/dt of expected value of another outcome; see *details*
d2EV[*depvar***|#]**	d2/dt2 of expected value of another outcome; see *details*
iEV[*depvar***|#]**	integral w.r.t time of the expected value of another outcome; see *details*
XB[*depvar***|#]**	expected value of the complex predictor of another outcome; see *details*
dXB[*depvar***|#]**	d/dt of the expected value of the complex predictor of another outcome; see *details*
d2XB[*depvar***|#]**	d2/dt2 of the expected value of the complex predictor of another outcome; see *details*
iXB[*depvar***|#]**	integral w.r.t time of the expected value of the complex predictor of another outcome; see *details*

More detailed descriptions can be found in **elements details**

An example

```
. list id time logb pro trt stime died if id==1 | id==2, noobs sepby(id)
```

id	time	logb	prothr~n	trt	stime	dpbc	dother
1	0	2.674149	12.2	D-penicil	1.09517	1	0
1	.525682	3.058707	11.2	D-penicil	.	.	.
2	0	.0953102	10.6	D-penicil	14.1523	0	0
2	.498302	-.2231435	11	D-penicil	.	.	.
2	.999343	0	11.6	D-penicil	.	.	.
2	2.10273	.6418539	10.6	D-penicil	.	.	.
2	4.90089	.9555114	11.3	D-penicil	.	.	.
2	5.88928	1.280934	11.5	D-penicil	.	.	.
2	6.88588	1.435084	11.5	D-penicil	.	.	.
2	7.8907	1.280934	11.5	D-penicil	.	.	.
2	8.83255	1.526056	11.5	D-penicil	.	.	.

Now I'm going to fit 12 different models...

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                ,                           /// options
                family(gaussian)            /// distribution
       )
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                ,                           /// options
                family(gaussian)            /// distribution
       )                                    ///
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
       )
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               , family(gamma)              /// distribution
       )                                    ///
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             //  vcv
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             /// vcv
       redistribution(t) df(5)              //  re dist.
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       (stime  trt                          /// response + covariate
              , family(rp, df(3)            /// distribution
                       failure(dother))     /// event indicator
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             /// vcv
       redistribution(t) df(5)              //  re dist.
```

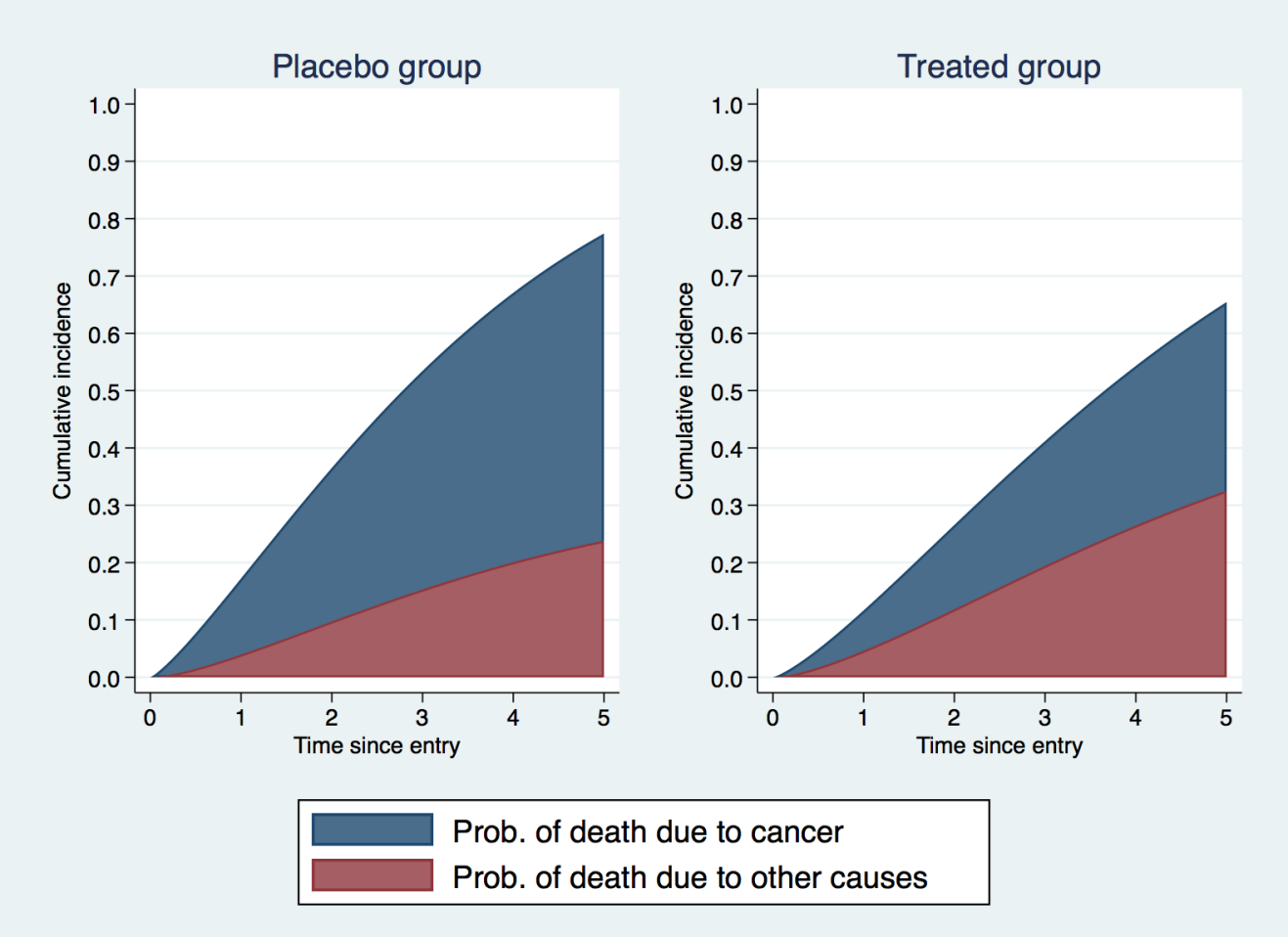
```
merlin (logb                                  /// log serum bilirubin
               time                           /// covariate
               time#trt                       /// interaction
               M1[id]@1                       /// random intercept
               time#M2[id]@1                  /// random slope
               ,                              /// options
               family(gaussian)               /// distribution
       )                                      ///
       (pro                                   /// prothrombin index
               rcs(time, df(3))               /// covariate
               M3[id]@1                       /// random effect
               , family(gamma)                /// distribution
       )                                      ///
       (stime  trt                            /// response + covariate
               dEV[logb] EV[pro]              /// associations
               , family(rp, df(3)             /// distribution
                       failure(dother))       /// event indicator
       )                                      ///
       ,                                      /// main options
       covariance(unstructured)               /// vcv
       redistribution(t) df(5)                //  re dist.
```

```
merlin (logb                                  /// log serum bilirubin
               time                           /// covariate
               time#trt                       /// interaction
               M1[id]@1                       /// random intercept
               time#M2[id]@1                  /// random slope
               ,                              /// options
               family(gaussian)               /// distribution
       )                                      ///
       (pro                                   /// prothrombin index
               rcs(time, df(3))               /// covariate
               M3[id]@1                       /// random effect
               , family(gamma)                /// distribution
       )                                      ///
       (stime  trt                            /// response + covariate
               trt#fp(stime, power(0))        /// tde
               dEV[logb] EV[pro]              /// associations
               , family(rp, df(3)             /// distribution
                       failure(dother))       /// event indicator
       )                                      ///
       ,                                      /// main options
       covariance(unstructured)               /// vcv
       redistribution(t) df(5)                //  re dist.
```

```
merlin (logb time time#trt M1[id]@1          /// model 1
               time#M2[id]@1 ,               ///
               family(gaussian)              ///
       )                                     ///
       (pro  rcs(time, df(3)) M3[id]@1       /// model 2
               , family(gamma)               ///
       )                                     ///
       (stime  trt                           ///
               trt#fp(stime, power(0))       /// model 3 - cause 1
               dEV[logb] EV[pro]             /// tde
               , family(rp, df(3)            /// distribution
                      failure(dother))       /// event indicator
       )                                     ///
       (stime  trt                           /// model 4 - cause 2
               trt#rcs(stime, df(3) log)     /// tde
               EV[logb] iEV[pro]             /// associations
               , family(weibull,             /// distribution
                       failure(dpbc))        /// event indicator
       )                                     ///
       ,                                     ///
       covariance(unstructured)
```

predictions

```
predict cif1, cif marginal outcome(3) at(trt 0)
predict cif1, cif marginal outcome(4) at(trt 0)
```



Joint model tutorials

There are a range of tutorials on my website (links are clickable)

- Joint longitudinal-survival model
- Joint longitudinal-survival model with time-dependent effects (non-proportional hazards)
- Weighted cumulative joint longitudinal-survival model
- Multivariate longitudinal and survival model
- Joint longitudinal and competing risks model
- Simulating from a joint longitudinal-survival model

Joint model tutorials

There are a range of tutorials on my website (links are clickable)

- Joint longitudinal-survival model
- Joint longitudinal-survival model with time-dependent effects (non-proportional hazards)
- Weighted cumulative joint longitudinal-survival model
- Multivariate longitudinal and survival model
- Joint longitudinal and competing risks model
- Simulating from a joint longitudinal-survival model

There's also a lot more...`merlin` can fit recurrent event models, joint frailty models, multi-state models

Variance joint modelling

Variance modelling

- Mixed effects models, and joint models have tended to focus on modelling the mean trajectory over time
- What if within patient variability is important, from a prognostic perspective?
- Blood pressure is the natural example - not only do mean trends play a role, but measurements of visit-to-visit variability, within an individual, could also be crucial (Barrett et al., 2019)

Mixed effects for the level 1 variance function

Hedeker et al. (2012), Goldstein et al. (2017) and Nordgren et al. (2020) proposed two-level models with complex level 1 variation, of the form,

$$
\begin{aligned}
y_{ij} &= X_{1ij}\beta_1 + Z_{1ij}b_{1j} + \epsilon_{ij} \\
\epsilon_{ij} &\sim N(0, \sigma_e^2) \\
\log(\sigma_e^2) &= X_{2ij}\beta_2 + Z_{2ij}b_{2j} \\
\begin{pmatrix} b_{1j} \\ b_{2j} \end{pmatrix} &\sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{b_1} & \\ \Sigma_{b_1 b_2} & \Sigma_{b_2} \end{pmatrix} \right]
\end{aligned}
$$

Linear mixed effects model

```
. merlin (logb time time#M2[id]@1 M1[id]@1, family(gaussian)), nolog
Fitting fixed effects model:
Fitting full model:
Mixed effects regression model                  Number of obs     =      1,945
Log likelihood = -1537.5928
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
logb:						
time	.1617427	.0128888	12.55	0.000	.1364812	.1870043
time#M2[id]	1	.	.	.	.	.
M1[id]	1	.	.	.	.	.
_cons	.5020114	.0593465	8.46	0.000	.3856943	.6183285
sd(resid.)	.3478019	.0067237			.3348702	.361233
id:						
sd(M1)	1.021243	.0431184			.9401342	1.109348
sd(M2)	.1717063	.0119214			.1498609	.1967362

Our own LME model

We can use merlin's utility functions to write our own evaluator:

```
real matrix gauss_logl(gml)
{
     y        = merlin_util_depvar(gml)          //dep. var.
     linpred  = merlin_util_xzb(gml)             //lin. pred.
     sdre     = exp(merlin_util_ap(gml,1))       //anc. param.
     return(lnnormalden(y,linpred,sdre))         //logl
}

merlin (logb time time#M2[id]@1 M1[id]@1, family(user, logl(gauss_logl) nap(1)))
```

Mixed effects for the level 1 variance function

We can fit this, and extend it, with `merlin`:

Log likelihood function

```
real matrix lev1(gml)
{
     y        = merlin_util_depvar(gml)                  //response
     linpred1 = merlin_util_xzb(gml)                     //lin. pred.
     varresid = exp(merlin_util_xzb_mod(gml,2))          //lev1 lin. pred
     return(lnnormalden(y,linpred,sqrt(varresid)))       //logl
}
```

Estimate the model

```
. merlin (logb time time#M2[id]@1 M1[id]@1, family(user, llf(lev1)))
         (M3[id]@1                        , family(null))
         , covariance(unstructured)
```

Mixed effects for the level 1 variance function

```
. merlin  (logb time#M2[id]@1 time M1[id]@1, family(user, llfunction(lev1_logl)))
>         (M3[id]@1, family(null))
>         , covariance(unstr) intmethod(gh) intpoints(7) nolog
Fitting fixed effects model:
Fitting full model:
Mixed effects regression model                  Number of obs     =      1,945
Log likelihood = -1528.5884
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
logb:						
time#M2[id]	1	.	.	.	.	.
time	.1357535	.0077394	17.54	0.000	.1205845	.1509225
M1[id]	1	.	.	.	.	.
_cons	.3784234	.0224853	16.83	0.000	.3343529	.4224938
null:						
M3[id]	1	.	.	.	.	.
_cons	-1.91191	.0533174	-35.86	0.000	-2.01641	-1.80741
id:						
sd(M1)	.7946658	.0183166			.7595647	.831389
sd(M2)	.0941229	.0054644			.0839996	.1054661
sd(M3)	.6975954	.0497115			.6066607	.8021606
corr(M2,M1)	.684333	.0338371			.6122327	.7451464
corr(M3,M1)	.5109115	.0616383			.3803594	.6214999
corr(M3,M2)	.6480568	.0576918			.5204771	.7473338

Mixed effects for the level 1 variance function

Let's make it easier:

```
merlin  (logb time#M2[id]@1 time M1[id]@1, family(gaussian, dapmodel(2)))
        (M3[id]@1                        , family(null))
        , covariance(unstr)
```

Mixed effects for the level 1 variance function

Let's make it easier:

```
merlin  (logb time#M2[id]@1 time M1[id]@1, family(gaussian, dapmodel(2)))
        (M3[id]@1                        , family(null))
        , covariance(unstr)
```

We can add anything we like:

- Further baseline covariates - `age trt`
- Non-linear effects:
  - `rcs(time, df(3))`
  - `fp(time, pow(1 2))`
  - `bs(time, order(4) df(3))`
- Random effects at higher levels - `M4[centre>id]@1`
- Assume another distribution - `family(poisson)`

Mixed effects for the level 1 variance function

- Remember we also had survival information
- Time-to-death should be incorporated in this setting (Barrett et al., 2019)

```
. merlin (logb time#M2[id]@1 time M1[id]@1, family(gaussian, dapmodel(2)))
>        (M3[id]@1, family(null))
>        (stime M1[id] M2[id] M3[id], family(weibull, failure(died)))
>        , covariance(unstr) intmethod(gh) intpoints(7) nolog
```

Mixed effects for the level 1 variance function

Mixed effects regression model — Number of obs = 1,945
Log likelihood = -1934.1523

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
logb:						
time#M2[id]	1	.	.	.	.	.
time	.1523727	.0056498	26.97	0.000	.1412992	.1634462
M1[id]	1	.	.	.	.	.
_cons	.3678922	.021518	17.10	0.000	.3257177	.4100668
null:						
M3[id]	1	.	.	.	.	.
_cons	-1.905143	.0541529	-35.18	0.000	-2.011281	-1.799005
stime:						
M1[id]	.7720355	.1758261	4.39	0.000	.4274227	1.116648
M2[id]	8.216058	2.406747	3.41	0.001	3.498921	12.9332
M3[id]	.2307741	.3783433	0.61	0.542	-.5107651	.9723133
_cons	-4.558101	.3183629	-14.32	0.000	-5.182081	-3.934121
log(gamma)	.5605197	.0754012	7.43	0.000	.4127361	.7083033
id:						
sd(M1)	.7882373	.0181291			.7534939	.8245827
sd(M2)	.1062894	.0046026			.0976406	.1157043
sd(M3)	.6948438	.0501899			.6031193	.8005181
corr(M2,M1)	.7316301	.0213419			.686981	.7707759
corr(M3,M1)	.5098832	.0619291			.3787083	.6209765
corr(M3,M2)	.6586738	.0561615			.5342262	.7551769

Mixed effects for the level 1 variance function

- I haven't changed the data structure for any of these examples
- We could add another biomarker
- We could add competing risks
- Predictions follow naturally,
  - fixed and marginal available, for any outcome
  - subject-specific - conditional on empirical Bayes estimates

Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models

Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models
- Through the complex linear predictor, we allow seamless development of novel models, and crucially, a way of making them immediately available to researchers through an accessible implementation

Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models
- Through the complex linear predictor, we allow seamless development of novel models, and crucially, a way of making them immediately available to researchers through an accessible implementation
- We are investigating the use of these methods in critical care settings, where monitoring is done at high frequencies

Future directions

- Model calibration and discrimination
- Timing of observations (Gasparini et al., 2020)

Future directions

- Model calibration and discrimination
- Timing of observations (Gasparini et al., 2020)
- Penalisation, cross-validation etc.

Future directions

- Model calibration and discrimination
- Timing of observations (Gasparini et al., 2020)
- Penalisation, cross-validation etc.
- It's general, and hence it can be slow(er) - I'm working on this

Future directions

- Model calibration and discrimination
- Timing of observations (Gasparini et al., 2020)
- Penalisation, cross-validation etc.
- It's general, and hence it can be slow(er) - I'm working on this
- For Bayesians...

  `morgana : merlin ...`

Future directions

- Model calibration and discrimination
- Timing of observations (Gasparini et al., 2020)
- Penalisation, cross-validation etc.
- It's general, and hence it can be slow(er) - I'm working on this
- For Bayesians...

  `morgana : merlin ...`
- Updates and tutorials here:
  www.mjcrowther.co.uk/software/merlin

References

Barrett, J., and L. Su. 2017. Dynamic predictions using flexible joint models of longitudinal and time-to-event data. *Statistics in Medicine* n/a–n/a. Sim.7209. URL `http://dx.doi.org/10.1002/sim.7209`.

Barrett, J. K., R. Huille, R. Parker, Y. Yano, and M. Griswold. 2019. Estimating the association between blood pressure variability and cardiovascular disease: An application using the ARIC Study. *Statistics in Medicine* 38(10): 1855–1868. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8074`.

Crowther, M. J. Extended multivariate generalised linear and non-linear mixed effects models .

———. 2020. merlin - A unified modeling framework for data analysis and methods development in Stata. *The Stata Journal* 20(4): 763–784. URL `https://doi.org/10.1177/1536867X20976311`.

Crowther, M. J., K. R. Abrams, P. C. Lambert, et al. 2013. Joint modeling of longitudinal and survival data. *Stata J* 13(1): 165–184.

Crowther, M. J., T. M.-L. Andersson, P. C. Lambert, K. R. Abrams, and K. Humphreys. 2016. Joint modelling of longitudinal and survival data: incorporating delayed entry and an assessment of model misspecification. *Statistics in medicine* 35(7): 1193–1209.

Gasparini, A., K. R. Abrams, J. K. Barrett, R. W. Major, M. J. Sweeting, N. J. Brunskill, and M. J. Crowther. 2020. Mixed-effects models for health care longitudinal data with an informative visiting process: A Monte Carlo simulation study. *Statistica Neerlandica* 74(1): 5–23. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/stan.12188`.

References 2

Goldstein, H., G. Leckie, C. Charlton, K. Tilling, and W. J. Browne. 2017. Multilevel growth curve models that incorporate a random coefficient model for the level 1 variance function. *Statistical methods in medical research* 962280217706728.

Hedeker, D., R. J. Mermelstein, and H. Demirtas. 2012. Modeling between-subject and within-subject variances in ecological momentary assessment data using mixed-effects location scale models. *Statistics in Medicine* 31(27): 3328–3336. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5338`.

Król, A., L. Ferrer, J.-P. Pignon, C. Proust-Lima, M. Ducreux, O. Bouché, S. Michiels, and V. Rondeau. 2016. Joint model for left-censored longitudinal data, recurrent events and terminal event: Predictive abilities of tumor burden for cancer evolution with application to the FFCD 2000–05 trial. *Biometrics* 72(3): 907–916.

Król, A., A. Mauguen, Y. Mazroui, A. Laurent, S. Michiels, and V. Rondeau. 2017. Tutorial in Joint Modeling and Prediction: a Statistical Software for Correlated Longitudinal Outcomes, Recurrent Events and a Terminal Event. *arXiv preprint arXiv:1701.03675* .

Li, N., R. M. Elashoff, and G. Li. 2009. Robust joint modeling of longitudinal measurements and competing risks failure time data. *Biom J* 51(1): 19–30. URL `http://dx.doi.org/10.1002/bimj.200810491`.

Lin, H., C. E. McCulloch, and S. T. Mayne. 2002. Maximum likelihood estimation in the joint analysis of time-to-event and multiple longitudinal variables. *Stat Med* 21(16): 2369–2382. URL `http://dx.doi.org/10.1002/sim.1179`.

References 3

Nordgren, R., D. Hedeker, G. Dunton, and C.-H. Yang. 2020. Extending the mixed-effects model to consider within-subject variance for Ecological Momentary Assessment data. *Statistics in Medicine* 39(5): 577–590. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8429`.

Philipson, P., I. Sousa, P. Diggle, P. Williamson, R. Kolamunnage-Dona, and R. Henderson. 2012. joineR - Joint Modelling of Repeated Measurements and Time-to-Event Data. URL `http://cran.r-project.org/web/packages/joineR/index.html`.

Rizopoulos, D. 2010. JM: An R Package for the Joint Modelling of Longitudinal and Time-to-Event Data. *J Stat Softw* 35(9): 1–33. URL `http://www.jstatsoft.org/v35/i09`.

———. 2015. JMbayes: joint modeling of longitudinal and time-to-event data under a Bayesian approach.

Rizopoulos, D., G. Verbeke, E. Lesaffre, and Y. Vanrenterghem. 2008. A Two-Part Joint Model for the Analysis of Survival and Longitudinal Binary Data with Excess Zeros. *Biometrics* 64(2): pp. 611–619. URL `http://www.jstor.org/stable/25502097`.