

Occupational exposure to beryllium in French industries

APPENDIX A

Bayesian Modelling

The log-transformed airborne measurements are denoted by “X” and the log-transformed urinary measurements are denoted by “Y”. The corresponding respective limits of quantification (on the log scale) are denoted by LOQ_X and LOQ_Y .

The airborne exposure measurements of worker i , X_{ij} are modelled for each SEG with a between-worker random effect model and a within-worker random effect representing day-to-day variability.

A similar model is used for the urinary data Y , with random effects for between-worker variability and within-worker variability.

We assume that, the by-subject log biological exposure measurements depend linearly on the by-subject (log-transformed) airborne exposure.

For the j th airborne observation of the i th worker, this can be written as:

$$X_{ij} \sim N(\mu_{SEG(i)}^X + \zeta_i^X, \sigma_W^X) \text{ with } \zeta_i^X \sim N(0, \sigma_B^X) \quad (1)$$

For the k th urinary observation of the i th worker, this can be written as:

$$Y_{ik} \sim N(\alpha^Y + \beta \cdot X_i + \xi_i^Y, \sigma_W^Y) \text{ with } \xi_i^Y \sim N(0, \sigma_B^Y) \quad (2)$$

where the by-subject-mean of the log-transformed exposure $X_i = \mu_{SEG(i)}^X + \zeta_i^X$

σ_B^X and σ_W^X are respectively the between-worker standard deviation and the within-worker standard deviation for the airborne log exposure measurements and $\mu_{SEG(i)}^X$ the population mean log airborne exposure of the SEG of subject i ;

σ_B^Y and σ_W^Y are respectively the between-worker standard deviation and the within-worker standard deviation for log urinary measurements, α^Y is the population log urinary exposure for a zero log airborne exposure (intercept), and β is the slope of the relation.

Censoring

In order to accommodate measurements below the LOQ, we introduce following notation for a truncated normal distribution $(\mu, \sigma)[a, b]$, a normal distribution truncated to the interval $[a, b]$. Thus, in our model corresponding to (1), a left censored value observed for j th observation of the i th worker would be specified as $X_{ij} \sim N(\mu_{SEG(i)}^X + \zeta_i^X, \sigma_W^X)[-\infty, LOQ_X]$. In our model corresponding to (2), a left censored value observed for k th observation of the i th worker would be specified as $Y_{ik} \sim N(\alpha^Y + \beta \cdot X_i + \xi_i^Y, \sigma_W^Y)[-\infty, LOQ_Y]$.

Prior distributions of parameters

Weakly informative prior distributions of the variability parameters σ_W^Y, σ_B^Y , are based on inverse gamma distribution with parameters (0.001, 0.001). It was chosen as it is the conditionally conjugate distribution for the inverse-chi2 distribution and leads to less computation time.

The prior distribution of the variability parameter σ_B^X is informative as information exists in the literature on this quantity. Following (McNally et al. 2014), we chose a lognormal with Geometric Mean GM=0.29 and Geometric Standard Deviation GSD=2.82 expressing that σ_B^X lies between 0.036 and 2.22 in 95% of the cases.

The prior distribution of the variability parameter σ_W^X is also informative as information exists in the literature on this quantity. Following (McNally et al. 2014), we chose a lognormal with Geometric Mean GM=0.92 and Geometric Standard Deviation GSD=1.64 expressing that σ_W^X lies between 0.34 and 2.38 in 95% of the cases.

Prior distributions of the parameters α^Y , β and μ_C^X are non-informative normal distributions.

Estimation of model parameters

The Bayesian model was fitted using Gibbs Sampling, a special case of Markov chain Monte Carlo (MCMC) methods (see (Wild et al. 1996) for Gibbs Sampling in the context of censored data, and (Lunn et al. 2012) for more recent general discussion of MCMC methods) that by default take the censored data into account. We used the freely available RJags software (JAGS 4.2.0 and R version 3.3.2). The code of the model used in Jags is detailed below.

We ran the Gibbs Sampling algorithm for N=50000 samples, a thinning of 5, and discarded the 10000 to account for “burn-in”. Convergence was checked by running three different chains starting from overdispersed starting points.

Jags codes for the Bayesian model

The data file contains:

- the constants nsubjects, nmeasures_uri, nmeasures_air and nSEG, containing respectively the number of subjects, the total number of urinary measurements, the total number of air exposure measurements, and the total number of SEG;
- the vectors subject_uri of size nmeasures_uri indicating for each urinary measurements to which subject it belongs;
- the vectors subject_air of size nmeasures_air indicating for each air exposure measurements to which subject it belongs;

- the vectors seg of size nsubjects indicating for each subjects to which SEG it belongs;
- the vectors xNA and yNA of size nmeasures_uri and nmeasures_air respectively, containing respectively the airborne and biological measurements. If below LOQ, the measurement contains NA;
- the vectors xLOQ and yLOQ of size nmeasures_uri and nmeasures_air respectively, containing 0 if measurements are observed, and the respective LOQ value if below LOQ;
- the vectors iscensoredx and iscensoredy of size nmeasures_uri and nmeasures_air respectively, containing 0 if measurements are below LOQ, and 1 if observed.

model

```
{
  for( i in 1 : nsubjects) {
    y[i] ~ dnorm(muy[i],tau.bety)
    muy[i]<- alphay + beta*x[i]
    x[i]~ dnorm(muxSEG[seg[i]] ,tau.betx)
  }
  for( j in 1 : nmeasures_air) {
    iscensoredx[j]~dinterval(xNA[j],xLOQ [j])
    xNA[j] ~ dnorm(x[subject_air[j]],tau.withx)
  }
  for( k in 1 : nmeasures_uri) {
    iscensoredy[k]~dinterval(yNA[k],yLOQ[k])
    yNA[k] ~ dnorm(y[subject_uri[k]],tau.withy)
  }

  #PRIORS
  for( h in 1 : nSEG ) {
    muxSEG[h]~ dnorm(0.0,1.0E-6)
  }

  tau.betx ~dlnorm(2.475, 0.4652) # GSD 1.33
  sigmabetx <- 1 / sqrt(tau.betx)
  # corresponding to sigmabetx ~LogN(GM=0.29; GSD=2.82 )

  tau.withx ~ dlnorm(0.167, 2.043) # GSD 2.5
```

```
sigmawithx <- 1 / sqrt(tau.withx)
# corresponding to sigmawithx ~LogN(GM=0.92; GSD=1.64 )
```

```
tau.withy ~ dgamma(0.0001,0.0001)
sigmawithy <- 1 / sqrt(tau.withy)
tau.bety ~ dgamma(0.001,0.001)
sigmabety<-1 / sqrt(tau.bety)
```

```
alphay ~ dnorm(0.0,1.0E-6)
beta ~ dnorm(0.0,1.0E-6)
```

```
# BLV calculation with French OEL =2 µg/m3
BLV<-exp(alphaypop + beta*log(2))
```

```
}
```