APPENDIX: A BAYESIAN APPROACH TO THE EVALUATION OF RISK-BASED MICROBIOLOGICAL CRITERIA FOR *CAMPYLOBACTER* IN BROILER MEAT

A.1. Computing the joint posterior density and the sampling of precision τ_w. For computation in OpenBUGS, we first write the joint posterior, assuming that individual measurements $(y_{ij''}, y_{1j'})$ are known in both data sets. The posterior for the bacteria concentration parameters is proportional to the full likelihood times the prior:

$$\underbrace{\prod_{j''}\Big(\Big(\prod_i N(y_{ij''} \mid \mu_{j''}, \tau_w)\Big)N(\mu_{j''} \mid \mu, \tau_b)\Big)}_{\text{Hansson et al.}} \times \underbrace{\prod_{j'}\Big(N(y_{1j'} \mid \mu_{j'}, \tau_w)N(\mu_{j'} \mid \mu, \tau_b)\Big)}_{\text{Lindblad et al.}}$$

$$\times \pi(\tau_w, \tau_b, \mu)$$

For the Lindblad et al. data, the BUGS code is straightforward, but in the data of Hansson et al. we have sample means $\bar{y}_{\cdot j''}$ and sample standard deviations $\mathrm{SD}(y_{\cdot j''})$. The likelihood function for the means can be simplified to $\prod_{j''} N(\bar{y}_{\cdot j''} \mid \mu_{j''}, \tau_w x_{j''})$ for BUGS coding. For parameter τ_w, we need to also include the likelihood contribution of the observed $\mathrm{SD}(y_{\cdot j''})$. Part of the likelihood function for τ_w comes from the data of Lindblad et al., where the single observations $y_{1j'}$ have distributions that depend on both variance components τ_w^{-1} and τ_b^{-1}, giving straightforward BUGS code. To also include evidence from SDs derived from the second data set, we need to obtain the full conditional distribution for τ_w from the data provided by Hansson et al., for constructing a Gibbs sampler, which can be implemented in BUGS by writing the corresponding full conditional distribution. From the joint posterior, we obtain the full conditional distribution for τ_w by further examining the part that involves the data of Hansson et al. and the prior $\pi(\tau_w)$:

$$\underset{\tau_w}{\propto} \prod_{i,j''} \tau_w^{0.5} e^{-0.5\tau_w (y_{ij''}-\mu_{j''})^2} \underbrace{\pi(\tau_w)}_{\propto \tau_w^{-1}} = \tau_w^{-1} \prod_{j''} \tau_w^{0.5 x_{j''}} e^{-0.5\tau_w \sum_i (y_{ij''}-\mu_{j''})^2}$$

$$= \tau_w^{0.5(\sum_{j''} x_{j''})-1} e^{-0.5\tau_w \sum_{i,j''}(y_{ij''}-\mu_{j''})^2}.$$

Then, using the well-known identity

$$\sum_{j''}\Big[\sum_i (y_{ij''}-\mu_{j''})^2\Big] = \sum_{j''}\Big[\sum_i (y_{ij''}-\bar{y}_{\cdot j''})^2 + x_{j''}(\bar{y}_{\cdot j''}-\mu_{j''})^2\Big],$$

we find that the full conditional distribution is proportional to

$$\tau_w^{0.5(\sum_{j''} x_{j''})-1} e^{-0.5\tau_w \sum_{j''}\Big[\sum_i (y_{ij''}-\bar{y}_{\cdot j''})^2 + x_{j''}(\bar{y}_{\cdot j''}-\mu_{j''})^2\Big]}$$

$$\propto \text{Gamma}\Big(0.5\sum_{j''} x_{j''}\ ,\ \ 0.5\sum_{j''}\Big[\sum_i (y_{ij''}-\bar{y}_{\cdot j''})^2 + x_{j''}(\bar{y}_{\cdot j''}-\mu_{j''})^2\Big]\Big).$$

Finally, using the sample standard deviation, we write $\sum_i (y_{ij''}-\bar{y}_{\cdot j''})^2 = (x_{j''}-1)\text{SD}(y_{\cdot j''})$. Then the full conditional distribution for τ_w can be expressed in BUGS as a Gamma distribution with parameters that can be computed from the sample means and sample standard deviations of the data of Hansson et al. and that depend on the unknown parameters $\mu_{j''}$, i.e. $\pi(\tau_w \mid \{\text{SD}(y_{\cdot j''}), \bar{y}_{\cdot j''}, \mu_{j''}\}_{j''=1,\ldots,J''}) = \text{Gamma}\Big(0.5\sum_{j''} x_{j''},\ 0.5\sum_{j''}\Big[(x_{j''}-1)\text{SD}(y_{\cdot j''}) + x_{j''}(\bar{y}_{\cdot j''}-\mu_{j''})^2\Big]\Big)$.

A.2. Computing illness probability. Using the posterior distribution of the model parameters based on the two data sets described above, we can make further predictions about the risk of illness in the servings produced. The predictions are based on a sequence of probability distributions; each of them depends on the output from the previous distribution. This chain forms a hierarchical model (i.e. the QMRA model) starting with the top-level parameters drawn from the posterior distribution describing the population of batches and carcasses. For the predictions of consumer risk, additional assumptions are needed as we have no corresponding data for the target population (Swedish consumers in this example) nor for all processing steps beyond the carcass level. Therefore, the predictions represent QMRA-model-dependent scenarios. The QMRA model was adapted from the existing model structure provided by Nauta et al. (2012).

The QMRA model begins with the conditional distribution for the batch mean $\mu_j \sim \text{N}(\mu, \sigma_b^{-2})$ and the conditional distribution for the $\log_{10}$ concentration of a contaminated carcass skin $y_c \sim \text{N}(\mu_j, \sigma_w^{-2})$. Then, according to Nauta et al. (2012) and EFSA (2011), the resulting concentration (cfu/g) in broiler meat is given as 10^{y_c-1}. The one-logarithm difference between the carcass and meat concentrations reflects an expert opinion [EFSA (2011)]. The number of bacteria in a serving corresponding to a portion of fresh chicken meat weighting w grams would then be $n_c \sim \text{Poisson}(w10^{y_c-1})$. For the serving weights w, we adopt a log-normal distribution with a mean of 189 grams and a variance of 127, which are based on Danish data for young adult males [Nauta et al. (2012)]. The bacteria would not survive in the broiler meat with proper cooking, but cross-contamination from fresh meat to salad could occur. Therefore, the exposure is assumed to occur via the type of salad servings that are prepared using similar hygienic practices as those in this example. Hence, the final number of bacteria (dose in a serving) is $d \sim \mathrm{Bin}(n_c, r)$, where r is the bacteria transfer probability when preparing a salad. In turn, this parameter was predicted from the posterior predictive distribution $\pi(r \mid r_1, \ldots, r_{55})$, where r_s, $s = 1, \ldots, 55$ are the observed transfer proportions in the 55 experimental preparations described in Nauta et al. (2008, 2012). Deviating slightly from Nauta et al. (2012), we modeled these data using a Bayesian hierarchical model: $r_s \sim \mathrm{Beta}(2, \beta)$ and $\beta \sim \mathrm{U}(1, 10^4)$. This sub-model accounts for the uncertainty of a limited empirical sample. Consequently, values below the observed minimum and above the observed maximum transfer proportion are also possible in the predictions. Finally, the dose response, conditional on d, is assumed to be given with parameters $\alpha_d = 0.145, \beta_d = 7.59$ [Teunis and Havelaar (2000)], so the probability of illness from eating a salad resulting from food preparation involving a contaminated broiler is

$$P_0(\text{ill} \mid d) = 0.33\Big(1 - \frac{\Gamma(\alpha_d + \beta_d)\Gamma(\beta_d + d)}{\Gamma(\beta_d)\Gamma(\alpha_d + \beta_d + d)}\Big). \tag{15}$$

The overall probability of illness, conditional on d, q, α, would be

$$P(\text{ill} \mid d, q, \alpha) = P(\text{contaminated broiler})P_0(\text{ill} \mid d) = q\,\underbrace{\alpha/(\alpha+2)}_{E(p_j|\alpha)}\,P_0(\text{ill} \mid d). \tag{16}$$

To obtain the probability of illness without conditioning on d, but conditional on the batch mean μ_j, the within-batch variance σ_w^2, and the QMRA model for the path from the carcass skin to servings, we compute $P(\text{ill} \mid q, \alpha, \mu_j, \sigma_w) = \frac{q\alpha}{\alpha+2} P_0(\text{ill} \mid \mu_j, \sigma_w)$ as

$$\frac{q\alpha}{\alpha+2} E_d(P_0(\text{ill} \mid d)) = \frac{q\alpha}{\alpha+2} \sum_{d=0}^{\infty} P_0(\text{ill} \mid d)\pi(d \mid \mu_j, \sigma_w). \tag{17}$$

The dose distribution $\pi(d \mid \mu_j, \sigma_w)$ is a predictive distribution given μ_j, σ_w and the QMRA model. The expected value can be computed approximately within every MCMC iteration step by

$$E_d(P_0(\text{ill} \mid d)) \approx \frac{1}{M} \sum_{m=1}^{M} P_0(\text{ill} \mid d^{(m)}), \tag{18}$$

where $d^{(m)}$ are repeatedly Monte Carlo sampled, $m = 1, \ldots, M$, based on the parameter values of the current iteration step in MCMC. This 2D Monte Carlo (Monte Carlo integration within the MCMC simulation) results in the "average illness probability" over possible doses d for each set of underlying parameters drawn from their posterior distribution. The illness probability can be written (eq. 16) conditionally on dose d and parameters q and α, or (eq. 17) conditionally on parameters q and α and the underlying parameters remaining after integration over d. In the latter case, the 95% posterior credible interval for $P(\text{ill} \mid q, \alpha, \mu_j, \sigma_w)$ was $[0, 0.036]$, showing the uncertainty from the remaining unknown parameters. The final result of the Bayesian analysis for a probability is a single number $P(\text{ill} \mid \text{evidence})$, which no longer depends on any of the unknown parameters. The evidence consists of the two data sets described earlier and the QMRA model for the carcass-to-serving pathway. The final probability, which does not depend on unknown parameters, is obtained as the posterior mean of the parametric expression: $P(\text{ill} \mid \text{evidence}) = E(q\alpha/(\alpha+2)P_0(\text{ill} \mid \mu_j, \sigma_w) \mid \text{evidence}) = 0.0033$.

A.3. 2D Monte Carlo. The expressions needed for RR involve several integrations. To calculate the probability $P(\text{ill} \mid q, \mu, \sigma_w, \sigma_b, \alpha)$ in the denominator in equation (7) for RR, we can write it as follows:

$$\int_{\Theta_j} P(\text{ill} \mid \theta_j, q, \mu, \sigma_w, \sigma_b, \alpha)\pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha)d\theta_j$$

$$= \int_{\Theta_j} P(\text{ill} \mid \theta_j)\pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha)d\theta_j,$$

because illness is conditionally independent of $q, \mu, \sigma_w, \sigma_b, \alpha$, given the batch parameters $\theta_j = (I_j, p_j, \mu_j)$. The illness probability involves integration over serving-specific parameters θ_s (which include y_c, n_c for the broiler, w for the serving size, cross-contamination rate r for the salad making, and d for the dose). Conditionally, given the serving parameters θ_s, the illness probability is independent of θ_j. Hence, we obtain

$$= \int_{\Theta_j} I_j p_j \int_{\Theta_s} P_0(\text{ill} \mid \theta_s)\pi(\theta_s \mid \theta_j)d\theta_s \; \pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha)d\theta_j$$

$$= \int_{\Theta_j} I_j p_j \int_{\Theta_s} P_0(\text{ill} \mid d)\text{Bin}(d \mid n_c, r)\pi(r)\text{Poisson}(n_c \mid w10^{y_c - 1})\pi(w)\text{N}(y_c \mid \mu_j, \sigma_w^{-2})d\theta_s$$

$$\times \pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha)d\theta_j$$

$$= q\frac{\alpha}{\alpha+2}\int_{\mu_j}\int_{\Theta_s} P_0(\text{ill} \mid d)\text{Bin}(d \mid n_c, r)\pi(r)\text{Poisson}(n_c \mid w10^{y_c - 1})\pi(w)\text{N}(y_c \mid \mu_j, \sigma_w^{-2})d\theta_s$$

$$\times\pi(\mu_j \mid \mu, \sigma_b) d\mu_j,$$

where the last expression is due to the conditional independence of batch parameters I_j and p_j from the rest of the parameters, given $q, \mu, \sigma_w, \sigma_b, \alpha$. With the Monte Carlo approach, the inner integration over the serving variability can be approximated by taking the Monte Carlo sample of serving parameters for each value of the batch parameters θ_j in the outer integral. In turn, the outer integration over the batch variability can be approximated by taking a Monte Carlo sample of batch parameters for each value of the core parameters. This procedure gives a Monte Carlo approximation of the illness probability, $P(\text{ill} \mid q, \mu, \sigma_w, \sigma_b, \alpha)$, which is a function of the core parameters, as

$$\approx q\frac{\alpha}{\alpha+2}\frac{1}{L}\sum_{l=1}^{L}\frac{1}{M}\sum_{m=1}^{M} P_0(\text{ill} \mid \theta_s^{(m,l)}),$$

where $\theta_s^{(m,l)}$ represents the Monte Carlo draws for the serving-specific parameters within batches, sampled with the current values of $q, \mu, \sigma_w, \sigma_b, \alpha$ at each MCMC iteration step, so that $\theta_j^{(l)}$ is sampled first, then $\theta_s^{(m,l)}$ depending on each $\theta_j^{(l)}$. Effectively, at each MCMC iteration L batches and M servings within each of the L batches are simulated.

The denominator in equation (12) is $P(\text{MC met} \mid q, \mu, \sigma_w, \sigma_b, \alpha)$

$$= \int_{\Theta_j} P(\text{MC met} \mid I_j, p_j, \mu_j, \sigma_w)\text{Bern}(I_j \mid q)\text{Beta}(p_j \mid \alpha, 2)\text{N}(\mu_j \mid \mu, \sigma_b^{-2}) d\theta_j$$

$$= \int_{\Theta_j} [qP_0(\text{MC met} \mid p_j, \mu_b, \sigma_w) + (1-q)]\text{Beta}(p_j \mid \alpha, 2)\text{N}(\mu_j \mid \mu, \sigma_b^{-2}) d\theta_j.$$

MC is met with 100% probability when the batch is not contaminated, which occurs with probability $1-q$. Hence, $P_0(\text{MC met} \mid p_j, \mu_j, \sigma_w)$ is the probability that the MC is met for a *contaminated batch*. This probability is a complicated expression involving all possible numbers of contaminated carcasses in a sample (depends on within-batch prevalence p_j), and the allowed maximum number of them exceeding the cfu limit (depends on μ_j, σ_w). Without efficient function calls, this expression would need to be tediously written for each microbiological criterion. However, the Monte Carlo simulation can again be used for approximation:

$$\approx q\frac{1}{L}\sum_{l=1}^{L} 1_{\{\text{MC met}\}}(p_j^{(l)}, \mu_j^{(l)}, \sigma_w) + (1-q),$$

where we draw L Monte Carlo draws for batch parameters from $\pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha)$, and $1_{\{\text{MC met}\}}$ is an indicator variable denoting "MC met", so the average over the Monte Carlo sample is an approximation of the probability.

The numerator in equation (12) is $P(\text{ill}, \text{MC met} \mid q, \mu, \sigma_w, \sigma_b, \alpha)$

$$= \int_{\Theta_j} P(\text{ill}, \text{MC met} \mid \theta_j, q, \mu, \sigma_w, \sigma_b, \alpha) \pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha) d\theta_j,$$

and, considering that the illness probability refers to the same batch to which the MC status refers, then both illness and MC status are conditionally independent of $q, \mu, \sigma_w, \sigma_b, \alpha$, given the batch parameters θ_j. Moreover, because illness and MC status are conditionally independent of each other, given θ_j, we can write the following:

$$= \int_{\Theta_j} P(\text{ill} \mid \theta_j) P(\text{MC met} \mid \theta_j) \pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha) d\theta_j.$$

This expression corresponds to "$\text{Risk}_{\text{MS,c}}$" in Nauta et al. (2012). The Monte Carlo integration over batch and serving parameters can be done as follows:

$$= \int_{\Theta_j} I_j p_j \Big[\int_{\Theta_s} P_0(\text{ill} \mid \theta_s) \pi(\theta_s \mid \mu_j, \sigma_w) d\theta_s \Big] P(\text{MC met} \mid I_j, p_j, \mu_j, \sigma_w)$$

$$\times \pi(\theta_j \mid q, \mu, \sigma_w, \sigma_b, \alpha) d\theta_j$$

$$= \int_{\Theta_j} I_j p_j \Big[\int_{\Theta_s} P_0(\text{ill} \mid \theta_s) \pi(\theta_s \mid \mu_j, \sigma_w) d\theta_s \Big] P(\text{MC met} \mid I_j, p_j, \mu_j, \sigma_w)$$

$$\times \text{Bern}(I_j \mid q) \text{Beta}(p_j \mid \alpha, 2) \text{N}(\mu_j \mid \mu, \sigma_b^{-2}) d\theta_j$$

$$= q \int_{\Theta_j} p_j \Big[\int_{\Theta_s} P_0(\text{ill} \mid \theta_s) \pi(\theta_s \mid \mu_j, \sigma_w) d\theta_s \Big] P_0(\text{MC met} \mid p_j, \mu_j, \sigma_w)$$

$$\times \text{Beta}(p_j \mid \alpha, 2) \text{N}(\mu_j \mid \mu, \sigma_b^{-2}) d\theta_j$$

$$\approx q \frac{1}{L} \sum_{l=1}^{L} p_j^{(l)} \Big[\frac{1}{M} \sum_{m=1}^{M} P_0(\text{ill} \mid \theta_s^{(m,l)}) \Big] 1_{\{\text{MC met}\}}(p_j^{(l)}, \mu_b^{(l)}, \sigma_w).$$

A.4. Posterior mean of RR. By defining $\theta = (\mu, \sigma_w, \sigma_b, q, \alpha)$ as the underlying parameters, and by writing the posterior mean of $RR(\theta)$, we obtain

$$E_\theta(RR(\theta) \mid \text{data}) = E_\theta\left(\frac{P(\text{ill} \mid \theta, \text{MC met})}{P(\text{ill} \mid \theta)} \mid \text{data} \right)$$

$$\overset{\approx^{\perp}}{\approx} \frac{E_\theta\Big(P(\text{ill} \mid \theta, \text{MC met}) \mid \text{data}\Big)}{E_\theta\Big(P(\text{ill} \mid \theta) \mid \text{data}\Big)} = \frac{P(\text{ill} \mid \text{MC met, data})}{P(\text{ill} \mid \text{data})} = RPR,$$

because $P(\text{ill} \mid \theta, \text{MC met})$ and $P(\text{ill} \mid \theta)$ are almost completely independent in each case involving a chosen MC.

References.

Nauta, M.J., Sanaa, M. and Havelaar A.H. (2012). Risk based microbiological criteria for *Campylobacter* in broiler meat in the European Union. *International Journal of Food Microbiology* **158** 209-217.

EFSA Panel on Biological Hazards (BIOHAZ) (2011). Scientific Opinion on Campylobacter in broiler meat production: control options and performance objectives and/or targets at different stages of the food chain. *EFSA Journal* **9(4):2105**. [141 pp.]. doi:10.2903/j.efsa.2011.2105.

Nauta, M.J., Fischer, A.R.H., van Asselt, E.D., de Jong, A.E.I., Frewer, L.J. and de Jonge, R. (2008). Food safety in the domestic environment: the effect of consumer risk information on human disease risks. *Risk Analysis* **28 (1)** 179-192.

Teunis, P.F.M. and Havelaar A.H. (2000). The beta poisson dose-response model is not a single-hit model. *Risk Analysis.* **20 (4)** 513-520.

A.5. BUGS codes. To obtain OpenBUGS, see `http://www.openbugs.net`.

A.5.1. *BUGS code for data1.*

```
model{
# Lindblad et al. data:
for(i in 1:NBpos){ # number of pos batches
for(j in 1:SB[i]){ # number of samples in batch (here: 1)
# concentration data in pos samples:
# conversion of cfu, subtract 2:
logcfu2[i,j] <- (logcfu[i,j]-2)
logcfu2[i,j] ~ dnorm(Lmub[i],tau_w)
}
Lmub[i] ~ dnorm(mu,tau_b)
}
# pos/neg-data:
```

```
# probability to sample a positive carcass from a positive batch
NBpos ~  dbin(prpos,N); prpos <- q*1
N <- NBpos+NBneg

# prediction (for a positive carcass):
logcfupredpos ~ dnorm(mu,tau_tot)
cfupredpos <- pow(10,logcfupredpos)

#priors:
q ~ dunif(0,1)
mu ~ dnorm(0,0.001)
tau_tot <- 1/(var_b+var_w)
sigma_b <- sqrt(var_b); var_b <- 1/tau_b
sigma_w <- sqrt(var_w); var_w <- 1/tau_w
tau_b ~ dgamma(0.001,0.001)
tau_w ~ dgamma(0.001,0.001)
phi <- var_w/(var_b+var_w)
}
```

A.5.2. *BUGS code for data2.*

```
model{
# Hansson et al. data:
for(i in 1:Nbatches){
# sample means reported as data in each batch:
# conversion of cfu, add log10(4):
mlogcfu2[i] <- (mlogcfu[i]+0.60206)
mlogcfu2[i] ~ dnorm(Hmub[i],tau_sample[i])
tau_sample[i] <- 1/(var_w/n.carcass[i])
Hmub[i] ~ dnorm(mu,tau_b)
# sample SDs reported as data:
# expression to be used for parameter in full conditional for tau_w:
expression[i] <- (pos.carcass[i]-1)*sdlogcfu[i]+
                  pos.carcass[i]*pow(mlogcfu2[i]-Hmub[i],2)
pos.carcass[i] ~ dbin(pwithin[i],n.carcass[i])
}
tau_w ~ dgamma(alphatw,betatw) # full conditional for tau_w
alphatw <- 0.5*sum(pos.carcass[])
betatw <- 0.5*sum(expression[])
sigma_w <- 1/sqrt(tau_w); var_w <- 1/tau_w
# prediction (for a positive carcass):
logcfupredpos ~ dnorm(mu,tau_tot)
cfupredpos <- pow(10,logcfupredpos)

# priors:
mu ~ dnorm(0,0.001)
tau_tot <- 1/(var_b+var_w)
sigma_b <- sqrt(var_b); var_b <- 1/tau_b
tau_b ~ dgamma(0.001,0.001)
phi <- var_w/(var_b+var_w)
for(i in 1:Nbatches){pwithin[i] ~ dbeta(apw,2)}
apw ~  dunif(1,10000)
mpw <- apw/(apw+2)
}
```

A.5.3. *BUGS code for computing the risk for quotient RPR.*

```
model{
# Lindblad et al. data:
for(i in 1:NBpos){  # number of batches
for(j in 1:SB[i]){  # number of samples in batch (here: 1)
# concentration data in pos samples:
# conversion of cfu, subtract 2:
logcfu2[i,j] <- (logcfu[i,j]-2)
logcfu2[i,j] ~ dnorm(Lmub[i],tau_w)
}
Lmub[i] ~ dnorm(mu,tau_b)
}
# pos/neg-data:
# probability to sample a positive carcass from a positive batch
NBpos ~  dbin(prpos,N); prpos <- q*mpw
N <- NBpos+NBneg

# Hansson et al. data:
for(i in 1:Nbatches){
# sample means reported as data in each batch:
# conversion of cfu, add log10(4):
mlogcfu2[i] <- (mlogcfu[i]+0.60206)
mlogcfu2[i] ~ dnorm(Hmub[i],tau_sample[i])
tau_sample[i] <- 1/(var_w/pos.carcass[i])
Hmub[i] ~ dnorm(mu,tau_b)
# sample SDs reported as data:
# expression to be used for paramater in full conditional for tau_w:
expression[i] <- (pos.carcass[i]-1)*sdlogcfu[i]+
                  pos.carcass[i]*pow(mlogcfu2[i]-Hmub[i],2)
pos.carcass[i] ~ dbin(pwithin[i],n.carcass[i])
}
tau_w ~ dgamma(alphatw,betatw) # full conditional for tau_w
alphatw <- 0.5*(sum(pos.carcass[]))
betatw <- 0.5*(sum(expression[]))
sigma_w <- 1/sqrt(tau_w); var_w <- 1/tau_w

# prediction (for a positive carcass):
logcfupredpos ~ dnorm(mu,tau_tot)
cfupredpos <- pow(10,logcfupredpos)
# prediction (for any carcass):
cfupred <- cfupredpos*IB; IB ~ dbern(prpos)

# priors:
q ~ dunif(0,1)
mu ~ dnorm(0,0.001)
tau_tot <- 1/(var_b+var_w)
sigma_b <- sqrt(var_b); var_b <- 1/tau_b
tau_b ~ dgamma(0.001,0.001)
phi <- var_w/(var_b+var_w)
for(i in 1:Nbatches){pwithin[i] ~ dbeta(apw,2)}
apw ~  dunif(1,10000)
mpw <- apw/(apw+2)
#############################
# Modeling weight of a serving:
     m <- 189; s2 <- 127
     # parameters for log-normal distribution:
     wmean <- log(m)-0.5*log(1+s2/(m*m))
     wtau <- 1/(  log(1+s2/(m*m))  )
##############################
# Modeling probability of bacteria transfer in a salad making:
for(i in 1:55){
       ptr[i] <- pow(10,-minuslogptr[i])  # from salad experiment
       ptr[i] ~ dbeta(2,ptrb)
   }
   ptrb ~ dunif(1,10000)
```

```
##############################
# Model of the outcome of  MC-criteria for a single batch,
# conditionally on  I, pw, mub, mtau_w in that batch
# criteria "n=5,c=1,m=1000", met if "c<=1"
# for example: MCn <- 5; MCc <- 1; MCm <- 3

# batch variables: I, pw, mub ,
# probabilities of these are affected by observed MC status.

        I ~ dbern(q)
        mub ~ dnorm(mu,tau_b)
        pw ~ dbeta(apw,2)
        pmet  <- phi((MCm-mub)*sqrt(tau_w))  # P(MCm<1000)
        campycarcasses  ~ dbin(pw,MCn) # for a contaminated batch
        notmet ~ dbin(pnotmet,campycarcasses)
        pnotmet <- 1-pmet # simulate how many times MC not met
        pMCmet <- I*step(MCc-notmet)+(1-I )
           # defines when MC is met for this batch
        MC ~ dbern(pMCmet)
           # MC is the outcome that can be given as data for this batch

logcfub~ dnorm(mub,tau_w)
w ~ dlnorm(wmean,wtau)
lambda <- w*pow(10,logcfub-1)*ptrnew
d ~ dpois(lambda)
ptrnew~ dbeta(2,ptrb)
P.ill0 <- 0.33*(1-exp(loggam(a+b)+loggam(b+d)-loggam(b)-loggam(a+b+d)))
a <- 0.145; b <- 7.59   # assumed dose resp parameters

# Risk for a batch for which MC was (or not) observed
P.illbatch <- I*pw*P.ill0
}
```

A.5.4. *BUGS code for parametric RR.*

```
model{
# Lindblad et al. data:
for(i in 1:NBpos){  # number of batches
for(j in 1:SB[i]){  # number of samples in batch (here: 1)
# concentration data in pos samples:
# conversion of cfu, subtract 2:
```

```
logcfu2[i,j] <- (logcfu[i,j]-2)
logcfu2[i,j] ~ dnorm(Lmub[i],tau_w)
}
Lmub[i] ~ dnorm(mu,tau_b)
}
# pos/neg-data:
# probability to sample a positive carcass from a positive batch
NBpos ~  dbin(prpos,N); prpos <- q*mpw
N <- NBpos+NBneg

# Hansson et al. data:
for(i in 1:Nbatches){
# sample means reported as data in each batch:
# conversion of cfu, add log10(4):
mlogcfu2[i] <- (mlogcfu[i]+0.60206)
mlogcfu2[i] ~ dnorm(Hmub[i],tau_sample[i])
tau_sample[i] <- 1/(var_w/pos.carcass[i])
Hmub[i] ~ dnorm(mu,tau_b)
# sample SDs reported as data:
# expression to be used for parameter in full conditional for tau_w:
expression[i] <- (pos.carcass[i]-1)*sdlogcfu[i]+
                  pos.carcass[i]*pow(mlogcfu2[i]-Hmub[i],2)
pos.carcass[i] ~ dbin(pwithin[i],n.carcass[i])
}
tau_w ~ dgamma(alphatw,betatw) # full conditional for tau_w
alphatw <- 0.5*(sum(pos.carcass[]))
betatw <- 0.5*(sum(expression[]))
sigma_w <- 1/sqrt(tau_w); var_w <- 1/tau_w

# prediction (for a positive carcass):
logcfupredpos ~ dnorm(mu,tau_tot)
cfupredpos <- pow(10,logcfupredpos)
# prediction (for any carcass):
cfupred <- cfupredpos*IB; IB ~ dbern(prpos)

# priors:
q ~ dunif(0,1)
mu ~ dnorm(0,0.001)
tau_tot <- 1/(var_b+var_w)
sigma_b <- sqrt(var_b); var_b <- 1/tau_b
```

```
tau_b ~ dgamma(0.001,0.001)
phi <- var_w/(var_b+var_w)
for(i in 1:Nbatches){pwithin[i] ~ dbeta(apw,2)}
apw ~  dunif(1,10000)
mpw <- apw/(apw+2)
#############################
# Modeling weight of a serving:
     m <- 189; s2 <- 127
     # parameters for log-normal distribution:
     wmean <- log(m)-0.5*log(1+s2/(m*m))
     wtau <- 1/(  log(1+s2/(m*m))  )
##############################
# Modeling probability of bacteria to transfer in a salad making:
for(i in 1:55){
       ptr[i] <- pow(10,-minuslogptr[i]) # from salad experiment
       ptr[i] ~ dbeta(2,ptrb)
   }
   ptrb ~ dunif(1,10000)
#############################
# Model of the outcome of  MC-criteria for a single batch,
# conditionally on  I, pw, mub, mtau_w in that batch
# criteria "n=5,c=1,m=1000", met if "c<=1"
# for example: MCn <- 5; MCc <- 1; MCm <- 3

for(j in 1:repsB){ # integration over batch variables pw, mub

        mub[j] ~ dnorm(mu,tau_b)
        pw[j] ~ dbeta(apw,2)
        pmet[j]  <- phi((MCm-mub[j])*sqrt(tau_w))  # P(MCm<1000)
        campycarcasses[j] ~ dbin(pw[j],MCn); # for a contam. batch
        notmet[j] ~ dbin(pnotmet[j],campycarcasses[j])
        pnotmet[j] <- 1-pmet[j]
           # simulate how many times MC is not met
        MCmetconbatch[j] <- step(MCc-notmet[j])
           # MC is (or not) met for a contaminated batch

for(i in 1:repsS){ # integration over contam. servings from a batch
logcfus[j,i] ~ dnorm(mub[j],tau_w)
ws[j,i] ~ dlnorm(wmean,wtau)
lambda[j,i] <- ws[j,i]*pow(10,logcfus[j,i]-1)*ptrnew[j,i]
d[j,i] ~ dpois(lambda[j,i])
ptrnew[j,i] ~ dbeta(2,ptrb)
P.ill0[j,i] <- 0.33*(1-exp(loggam(a+b)+loggam(b+d[j,i])-loggam(b)-
                           loggam(a+b+d[j,i])))
}
#################################
# Calculating risk for a given contaminated batch:
P.ill[j] <- pw[j]*mean(P.ill0[j,])
# Calculating joint probability of 'risk' and 'MC met'
# for a contaminated batch:
P.illMCmet[j] <- step(MCc-notmet[j])*pw[j]*mean(P.ill0[j,])
        }
a <- 0.145; b <- 7.59   # assumed dose resp parameters
# calculating mean risk:
mP.ill <- q*mean(P.ill[])

# calculating MRRR:
mP.illMCmet <- q*mean(P.illMCmet[])
mrrr <- mP.illMCmet/mP.ill
#=P(ill & MCmet | population parameters)/P(ill | population parameters)
# calculating RR:
mP.illgivenMCmet <- mP.illMCmet/PMCmet
PMCmet <- q*mean(MCmetconbatch[])+(1-q)
rr <- mP.illgivenMCmet/mP.ill
#=P(ill | MCmet, population parameters)/P(ill | population parameters)
}
```

A.5.5. *Data.*

```
NBpos=88
NBneg=529
SB=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
logcfu=structure(.Data=c(
2.60,2.60,2.60,2.60,2.60,2.60,2.60,2.60,2.90,2.90,2.90,3.08,
3.08,3.20,3.26,3.34,3.40,3.41,3.51,3.51,3.56,3.56,3.60,3.64,
3.64,3.64,3.68,3.88,3.88,3.92,3.92,3.94,3.94,3.94,3.94,3.96,
3.97,4.01,4.02,4.02,4.06,4.07,4.09,4.10,4.16,4.16,4.18,4.19,
4.23,4.25,4.26,4.26,4.29,4.30,4.31,4.32,4.36,4.38,4.41,4.41,
4.50,4.60,4.60,4.64,4.65,4.68,4.70,4.71,4.71,4.76,4.77,4.82,
4.86,4.90,4.96,4.98,4.99,4.99,5.02,5.06,5.14,5.26,5.32,5.40,
5.42,5.42,6.17,7.15),.Dim=c(88,1))
Nbatches=20
mlogcfu=c(2.31,1.96,1.38,2.98,2.87,2.76,3.02,2.69,3.15,2.63,2.74,
2.32,2.62,2.62,1.35,1.21,2.19,1.39,2.13,2.11)
sdlogcfu=c(0.61,0.51,0.60,0.48,0.71,0.39,0.58,0.40,0.49,0.37,0.37,
0.26,0.49,0.35,0.81,0.80,0.48,0.75,0.69,0.61)
pos.carcass=c(24,10,21,16,12,13,5,10,20,11,20,15,17,17,17,18,23,20,19,11)
n.carcass=  c(25,10,23,16,12,13,5,10,20,11,20,15,17,17,20,21,23,20,20,11)
minuslogptr=c( 2.24, 2.36, 2.37, 2.58, 2.82, 2.86, 3.16,
3.17, 3.47, 3.52, 3.57, 3.83, 3.83, 3.84, 3.87, 3.89,
3.89, 3.90, 3.94, 4.03, 4.09, 4.42, 4.53, 4.54, 4.54,
4.62, 4.62, 4.68, 4.73, 4.76, 4.84, 4.92, 4.93, 4.95,
4.97, 5.20, 5.25, 5.27, 5.39, 5.47, 5.60, 5.83, 5.89,
5.95, 5.96, 6.02, 6.23, 6.38, 6.96, 7.37, 7.90, 8.20,
9.00, 9.00, 9.00)
```