water

MDPI

Article

Supplementary Data and Codes: Predictive Water Virology: Hierarchical Bayesian Modeling for Estimating Virus Inactivation Curve

Syun-suke Kadoya [1], Osamu Nishimura [1] Hiroyuki Kato [2] and Daisuke Sano [1, 3,*]

[1] Department of Civil and Environmental Engineering, Tohoku University, Aoba 6-6-06, Aramaki, Aoba-ku, Sendai, Miyagi 980-8579; kadotani9@gmail.com; osamu.nishimura.d2@tohoku.ac.jp;
[2] New Industry Creation Hatchery Center, Tohoku University, Aoba 6-6-10, Aramaki, Aoba-ku, Sendai, Miyagi 980-8579; h-katou@jiwet.or.jp;
[3] Department of Frontier Science for Advanced Environment, Graduate School of Environmental Studies, Tohoku University, Aoba 6-6-06, Aramaki, Aoba-ku, Sendai, Miyagi 980-8579, Japan; daisuke.sano.e1@tohoku.ac.jp;
* Correspondence: daisuke.sano.e1@tohoku.ac.jp; Tel.: +81-22-795-7481

Received: 30 August 2019; Accepted: 16 October 2019; Published: date

Table S1. Norovirus data analyzed for model construction.

No.	*k*	*m*	*n*	FC[1]	*k'*	pH	T	Water type	Assay[4]	Genotype	Experiment
1	0.99	0.40	1.29	0.500	0.19	7.2	22	C[2]	Genome	GII.4	1
2	0.87	1.91	0.57	0.500	0.19	7.2	22	C	Genome	GII.4	1
3	0.10	1.10	0.79	1.420	0.12	8.0	24	C	Infectivity	MNV	2
4	0.22	0.81	0.10	1.760	0.20	8.0	24	C	Infectivity	MNV	2
5	1.23	0.50	0.10	0.191	0.38	7.2	5	P[3]	Genome	GII.4	3
6	2.00	0.63	0.71	0.193	0.38	7.2	5	P	Genome	MNV	3
7	0.27	1.20	0.18	0.193	0.38	7.2	5	P	Genome	MNV	3
8	22.12	0.48	1.28	0.193	0.38	7.2	5	P	Infectivity	MNV	3
9	0.96	0.40	0.10	0.184	0.07	7.2	20	P	Genome	MNV	3
10	0.37	0.76	0.11	0.184	0.07	7.2	20	P	Genome	MNV	3
11	5.00	0.40	0.33	0.184	0.07	7.2	20	P	Infectivity	MNV	3
12	1.95	0.72	0.10	1.000	4.66	7.0	5	C	Infectivity	MNV	4

[1]initial concentration of free chlorine [ppm]
[2]contaminated water (converted to 1)
[3]purified water (converted to 0)
[4]measuring assay for virus concentration (Infectivity: 0, Genome: 1)

Code S1. R code for model construction

```
library(rstan)
library(GGally)
library(reshape2)

d <- read.csv("norofc.csv", header = TRUE)      #Input the source data file like Table 1
N <- nrow(d)      #Enumeration of the number of data
S <- 2          #Two genotype used here (GII.4 and MNV)
```

```
E <- 4          #Datasets were derived from four articles

data_k <- list(N=N, S=S, E=E, Y=log(d$k), pH=d$pH, T=d$T,
               Assay=d$Assay, Wqual=d$Wqual, Experiment=d$Experiment,
                Strain=d$Strain
                )
data_m <- list(N=N, S=S, E=E, Y=log(d$m), pH=d$pH, T=d$T,
               Assay=d$Assay, Wqual=d$Wqual, Experiment=d$Experiment,
                Strain=d$Strain
                )
data_n <- list(N=N, S=S, E=E, Y=log(d$n), pH=d$pH, T=d$T,
               Assay=d$Assay, Wqual=d$Wqual, Experiment=d$Experiment,
                Strain=d$Strain
                )

fit_k <- stan(file="noro_k.stan", data=data_k, seed=12345)
                #Run stan code for hierarchical Bayesian modeling for k
fit_m <- stan(file="noro_m.stan", data=data_m, seed=12345)
                #Run stan code for hierarchical Bayesian modeling for m
fit_n <- stan(file="noro_n.stan", data=data_n, seed=12345)
                #Run stan code for hierarchical Bayesian modeling for n
```

Code S2. Stan code for model construction

```
data {                                  #Declaration of variables in 'data_k (or n, m)' in R
  int N;
  int S;
  int E;
  real Y[N];
  real pH[N];
  real T[N];
  real Wqual[N];
  real Assay[N];
  int<lower=1, upper=E> Experiment[N];
  int<lower=1, upper=S> Strain[N];
}

parameters {                            #Declaration of coefficients of each variable
  real a0;                              #Putative real coefficients ('x0')
  real b0;
  real c0;
  real d0;
  real e0;
  real aS[Strain];                      #Coefficients specific to each genotype('xS')
  real bS[Strain];
  real cS[Strain];
  real dS[Strain];
  real eS[Strain];
  real a[Experiment];                   #Coefficients specific to each experiment('x')
  real b[Experiment];
  real c[Experiment];
  real d[Experiment];
```

```
    real e[Experiment];
    real <lower=0> sigma;                    #Standard deviation of the used distribution
    vector<lower=0>[S] s_aS;    #Standard deviation generating genotype dependent sensitivity
    vector<lower=0>[S] s_bS;
    vector<lower=0>[S] s_cS;
    vector<lower=0>[S] s_dS;
    vector<lower=0>[S] s_eS;
    vector<lower=0>[E] s_a;     #Standard deviation generating differences among experiments
    vector<lower=0>[E] s_b;
    vector<lower=0>[E] s_c;
    vector<lower=0>[E] s_d;
    vector<lower=0>[E] s_e;
}

transformed parameters {        #Expression of population parameters for used distribution by
                                  using water-quality
    real mu[N];                 #Declaration of the population parameter μ (normal distribution)
    for(n in 1:N){
            mu[n] = a[Experiment[n]]+ b[Experiment[n]]*pH[n]+ c[Experiment[n]]*T[n]+
                d[Experiment[n]]*Assay[n] + e[Experiment[n]]*Wqual[n];
            }
}

model{

    for (t in 1:Strain){   #Genotype-dependent values are generated from normal distributions,
                             which is putatively common among all genotypes
      aS[t] ~ normal(a0, s_aS);
      bS[t] ~ normal(b0, s_bS);
      cS[t] ~ normal(c0, s_cS);
      dS[t] ~ normal(d0, s_dS);
      eS[t] ~ normal(e0, s_eS);
      s_aS[t] ~ gamma(10, 10);          #Prior distribution for standard deviation providing
                                          differences among genotypes
      s_bS[t] ~ gamma(10, 10);
      s_cS[t] ~ gamma(10, 10);
      s_dS[t] ~ gamma(10, 10);
      s_eS[t] ~ gamma(10, 10);
    }

  for (x in 1:E){  #Observed values are generated from normal distributions specific to each
                     genotype
      a[x] ~ normal(aS[Strain[x]], s_a);
      c[x] ~ normal(bS[Strain[x]], s_b);
      d[x] ~ normal(cS[Strain[x]], s_c);
      e[x] ~ normal(dS[Strain[x]], s_d);
      f[x] ~ normal(eS[Strain[x]], s_e);
      s_a[x] ~ gamma(10, 10);           #Prior distribution for standard deviation providing
                                          differences among disinfection tests
      s_b[x] ~ gamma(10, 10);
      s_c[x] ~ gamma(10, 10);
      s_d[x] ~ gamma(10, 10);
```

```
            s_e[x] ~ gamma(10, 10);
        }

        a0 ~ normal(14, 5);          #Prior distributions for putatively real coefficients.
                                       It is better for you to start following condition: normal (0, 10)
        b0 ~ normal(-12, 5);
        c0 ~ normal(-0.5, 0.1);
        d0 ~ normal(-1, 1);
        e0 ~ normal(-1, 0.5);

        for(i in 1:N)
            Y[i] ~ normal(mu[i], sigma);
    }

    generated quantities {              #Generation of predictive values by constructed models.
      vector[N] y_rep;
      vector[N] log_lik;
      for(n in 1:N){
        y_rep[n] = normal_rng(mu[n], sigma);
        log_lik[n] = normal_lpdf(Y[n]| mu[n], sigma);
      }
}
```

Code S3. R code for the prediction of EFH model parameters

```
ms_k <- rstan::extract(fit_k)       #Extract the estimated coefficients for EFH model parameter
ms_m <- rstan::extract(fit_m)
ms_n <- rstan::extract(fit_n)

####Example (pH=6.74, T = 16.6, Assay = genome, Water type = contaminated, GII strain) ####
param_k <- ms_k$aS[1] +ms_k$bS[1]*6.74/7 +ms_k$cS[1]*16.6/20 +
           ms_k$dS[1]*1 +ms_k$eS[1]*1
param_m <- ms_m$aS[1] + ms_m$bS[1]*6.74/7 + ms_m$cS[1]*16.6/20 +
           ms_m$dS[1]*1 + ms_m$eS[1]*1
param_n <- ms_n$aS[1] +ms_n$bS[1]*6.74/7 +ms_n$cS[1]*16.6/20 +
           ms_n$dS[1]*1 +ms_n$eS[1]*1

k_pre <- rnorm(1000, param_k, ms_k$sigma)       #Regeneration of predictive parameters
m_pre <- rnorm(1000, param_m, ms_m$sigma)
n_pre <- rnorm(1000, param_n, ms_n$sigma)

exp(quantile(k_pre, c(0.025, 0.25, 0.5, 0.75, 0.975)))    #Extraction of 95% confidence intervals
exp(quantile(m_pre, c(0.025, 0.25, 0.5, 0.75, 0.975)))
exp(quantile(n_pre, c(0.025, 0.25, 0.5, 0.75, 0.975)))

####Example of predictive inactivation curve####
C <- 0.5                            #Initial concentration of free chlorine [ppm]
t <- c(0, 1, 2, 3, 4, 5)            #contact time [min]
k1 <- 0.6                           #Decay constant of free chlorine [min-1]

EFH_0.025 <- -k_0.025*C^n_0.025*t^m_0.025 * ((1-exp(-n_0.025*k1*t/m_0.025))
                                             /(n_0.025*k1*t/m_0.025))^m_0.025
```

```
EFH_0.5 <- -k_0.5*C^n_0.5*t^m_0.5 * ((1-exp(-n_0.5*k1*t/m_0.5))/(n_0.5*k1*t/m_0.5))^m_0.5
EFH_0.975 <- -k_0.975*C^n_0.975*t^m_0.975 * ((1-exp(-n_0.975*k1*t/m_0.975))
                                              /(n_0.975*k1*t/m_0.975))^m_0.975

EFH <- data.frame(t, EFH_0.025, EFH_0.5, EFH_0.975)
EFH[is.na(EFH)] = 0                          #NA is converted to zero

library(ggplot2)

plot <- ggplot() + geom_line(data=a, aes(x=t, y=EFH_0.025), color="red", size=1.5) +
                   geom_line(data=a, aes(x=t, y=EFH_0.5), color="blue", size=1.5) +
                   geom_line(data=a, aes(x=t, y=EFH_0.975), color="green", size=1.5)
quartz()                                     # for Mac
plot                                         #draw the predictive curves in quartz window
```

Author Contributions: Conceptualization, S. Kadoya and D. Sano; Methodology, S. Kadoya.; Validation, O. Nishimura and D. Sano; Formal Analysis, S. Kadoya; Investigation, S. Kadoya; Data Curation, S. Kadoya; Writing-Original Draft Preparation, S. Kadoya.; Writing-Review & Editing, D. Sano and O. Nishimura; Visualization, S. Kadoya.; Supervision, D. Sano; Project Administration, D. Sano.

Funding: This study was supported by Gesuido Academic Incubation to Advanced Project, Japan Ministry of Land, Infrastructure, Transport and Tourism, and "The Sanitation Value Chain: Designing Sanitation Systems as Eco-Community Value System" Project, Research Institute for Humanity and Nature (RIHN, Project No.14200107).

Conflicts of Interest: We do not have any conflicts of interest in this study.

References

1. WHO. Sanitation safety planning: manual for safe use and disposal of wastewater, greywater and excreta. Geneva, World Health Organization, **2015**.